AD-A118 899 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/7
AN IMPROVED TECHNIQUE FOR ALTITUDE TRACKING OF AIRCRAFT.(U)
MAR 81 J W ANDREWS F19628-80-C-0002
UNCLASSIFIED ATC-105 FAA-RD-82-14 NL

1 OF 1
AD
A118899

END
DATE
FILMED
10 82
DTIC

Technical Report Documentation Page

1. Report No.
FAA-RD-82-14

2. Government Accession No.
AD-A118 899

3. Recipient's Catalog No.

4. Title and Subtitle
An Improved Technique for Altitude Tracking of Aircraft

5. Report Date
25 March 1981

6. Performing Organization Code

7. Author(s)
J.W. Andrews

8. Performing Organization Report No.
ATC-105

9. Performing Organization Name and Address
Massachusetts Institute of Technology
Lincoln Laboratory
P.O. Box 73
Lexington, MA 02173

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DOT-FA77WAI-817

12. Sponsoring Agency Name and Address
Department of Transportation
Federal Aviation Administration
Systems Research and Development Service
Washington, DC 20591

13. Type of Report and Period Covered
Project Report

14. Sponsoring Agency Code

15. Supplementary Notes
The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology under Air Force Contract F19628-80-C-0002.

16. Abstract

When simple linear recursive tracking techniques are applied to quantized altitude reports, certain errors in estimation of altitude and altitude rate can be attributed to the response of the tracker to transitions between quantization levels. These errors can be reduced by use of an estimation technique which explicitly recognizes the quantized nature of the inputs. Smoothing of the level occupancy time (i.e., the time spent at each quantization level) can be used to control the response to redundant samples taken at the same quantization level. Further improvement is achieved by consistency tests which use particular properties of quantized data to detect changes in rate. This document presents a theoretical analysis of tracker response to quantized inputs. A tracking algorithm is synthesized using these techniques and simulation results using various altitude profiles are presented.

17. Key Words
Collision Avoidance
Quantization
BCAS
Algorithm
Tracking
Altimetry

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	64	

CONTENTS

CONTENTS (Cont'd)

ILLUSTRATIONS

ILLUSTRATIONS

1.0 INTRODUCTION

The ability of the FAA Beacon Collision Avoidance System (BCAS) to detect and properly respond to collision hazards depends upon the accurate estimation of aircraft vertical rates. These estimates are based upon altitude reports derived from encoding barometric altimeters and are quantized in 100-foot altitude increments.

In early BCAS testing it was noted that the response of the altitude tracking algorithm to a single isolated transition from one quantization level to the next was a rate estimate of substantial magnitude. Since isolated transitions often occur when the actual rate is negligible, the resulting error in rate estimation could lead to improper selection of avoidance maneuver directions.

Further study revealed that a large part of the problem was due to the manner in which the linear recursive "alpha-beta" tracking algorithm responded to quantized data inputs. Reductions in tracking gain* reduced tracker response to isolated altitude transitions. Although such gain reductions eliminated the over-response problem, they also substantially reduced the capability of the tracker to respond to actual rate changes. As a consequence, Lincoln Laboratory undertook an investigation of alternative tracking techniques which suppress response to isolated altitude quantization transitions while responding promptly to actual rate changes. This effort yielded improved tracking algorithms for use in collision avoidance systems. It also provided theoretical insight into the general problem of rate tracking with coarsely quantized inputs.

As employed in this document the term "coarsely quantized" refers to a system in which the rates of interest are such that at most one quantization level is crossed between samples. For the 100-foot quantization levels and 1-Hertz update rate of BCAS, all rates below the nominal design limit of 6000 FPM result in coarsely quantized rate tracking behavior. If altitude tracking is based upon ground-based sensor data, the update interval may be 4 seconds or greater. Higher altitude rates may then result in several quantization levels being crossed between samples and the ability of the tracker to estimate rate is limited more by the sampling rate than by the altitude quantization. With 4-second update intervals it is more appropriate to refer to the tracking process as "coarsely sampled". The design principles used in the coarsely quantized tracker apply with only slight modification to the coarsely sampled regime, as will be described in Section 8.0.

---

*Broste, Nels A., "A Vertical Tracker Redesign for Active BCAS," MTR-79W00431, The MITRE Corporation (March 1980).

2.0 MODEL OF THE ALTITUDE TRACKING PROCESS

The design of estimation algorithms requires that a mathematical model of the measurement process be defined in order to provide the basis for both mathematical analysis and computer simulation efforts. This section discusses those characteristics of air traffic and altimetry which are relevant to the selection of an appropriate model

2.1 Error Characteristics of Altitude Reports

Observation of barometric altimetry data indicates that errors in pressure transducer output do not vary significantly from sample to sample. Such constant errors can be viewed as biases which contribute errors in position determination but do not influence the accuracy of altitude rate estimation. Hence minimization of error due to sample-to-sample jitter is not a primary problem in altitude tracking (although tracking should be designed to smooth such errors whenever they occur). The principal challenge of altitude tracking is to produce accurate estimates in the presence of measurement quantization and finite sampling rate.

2.2 Relative Significance of Rate and Position Errors

In collision avoidance applications, the quantity of interest is the aircraft position at some future time (such as the time of closest approach or the time at which a response to collision avoidance instructions would begin). Altitude projections are obtained by a *simple linear projection* of aircraft motion according to

$$\hat{z}_t = \hat{z}_0 + \hat{\dot{z}}_0 t \qquad (2.1)$$

where $\hat{z}_t$ is the projected altitude at time t and $\hat{z}_0$, $\hat{\dot{z}}_0$ are the current estimated position and rate. The rate is assumed to be constant for the duration of the projection. If the errors in these quantities are denoted by $e_{\hat{z}_t}$, $e_{\hat{z}_0}$ and $e_{\hat{\dot{z}}_0}$ then

$$e_{\hat{z}_t} = e_{\hat{z}_0} + e_{\hat{\dot{z}}_0} t \qquad (2.2)$$

It can be seen that the longer the projection time, the greater is the significance of the rate error in comparision to the position error. For 30 seconds projection time, a rate error of 200 FPM is equivalent (in terms of projection error) to a position error of 100 feet.

Quantized altitude measurements include a quantization error which lies between $-q/2$ and $q/2$. Any tracking which goes beyond mere use of the raw report involves attempting to determine exactly where within the quantization interval (of width q) the actual position lies. Hence for $q = 100$ feet the maximum reduction in projection error which can be achieved by improved position estimation is less than or equal to 50 feet. It can now be seen that position estimation errors are unlikely to be a critical issue in the selection of an altitude tracking technique. This is true first because errors on the order of 50 feet are not large enough to significantly affect the performance of collision avoidance systems. Secondly, experience indicates that the tracking design cannot be expected to significantly reduce altitude determination errors below $\pm q/2$.

Rate estimation errors however can be critical to resolution success and can vary significantly depending upon tracking technique. An error of 1000 FPM projected over a 30 second interval produces an error in projected altitude of 500 feet. Such errors are comparable in magnitude to the amount of altitude deviation which can be effected by resolution commands. Hence, rate errors can and do influence the success of resolution.

2.3 Characteristics of Aircraft Trajectories

Several aspects of tracking algorithm design are influenced by the types of altitude trajectories which are to be expected in operation. The following paragraphs discuss some of the significant characteristics of altitude trajectories.

Pilots generally attempt to maintain a near-zero altitude rate at the desired flight altitude or to hold a constant non-zero rate in transitioning between altitudes. In some cases larger altitude changes occur in a number of steps as air traffic control issues clearances to successive altitude levels. Typical aircraft trajectories might be envisioned as a series of constant rate segments with periods of acceleration occurring whenever the rate changes. In reality of course, aircraft never maintain exactly constant rates. And even when attempting to hold constant altitude there is some oscillation about the desired altitude. Hence isolated transitions between adjacent quantization levels may be observed even for aircraft in nominally level flight.

Lower performance aircraft tend to climb and descend at rates between 500 FPM and 1000 FPM. Higher performance aircraft, such as jet transports, use higher rates, but seldom exceed 6000 FPM. The BCAS system is designed to provide nominal performance for rates up to 6000 FPM.

In going from one altitude rate to another, aircraft are usually assumed to accelerate at values between 0.10g and 0.25g. This level of acceleration allows the rate to change substantially between quantization levels. Consider for instance an aircraft which accelerates with constant acceleration from an initial vertical rate $\dot{z}_0$ to a final rate $\dot{z}_f$. The altitude is given by:

$$z(t) = \begin{cases} z_0 + \dot{z}_0 t + \dfrac{a}{2} t^2 & 0 < t < \dfrac{\dot{z}_f - \dot{z}_0}{a} \\[2ex] z_0 - \dfrac{(\dot{z}_f - \dot{z}_0)^2}{2a} + \dot{z}_f t & t > \dfrac{\dot{z}_f - \dot{z}_0}{a} \end{cases} \tag{2.3}$$

where a is the acceleration and z_0 is the initial altitude. At the time the final rate is achieved the aircraft position is

$$z_0 + \frac{\dot{z}_f^2 - \dot{z}_0^2}{2a} \tag{2.4}$$

It can be seen from the above expression that an aircraft which initiates a 0.25g vertical acceleration from level flight can achieve a rate of more than 2400 FPM before moving 100 feet vertically. This indicates that the coarseness of the altitude quantization is insufficient for accurate rate tracking during periods of acceleration. However it should also be noted that periods of acceleration do not persist for more than a few seconds. At 0.25g for instance, the rate of 2400 FPM is achieved from level flight in only 5.0 seconds.

2.4 System Model

Figure 2.1 is a diagram of the system model which will be used in the following investigation of altitude tracking. In this diagram the Laplacian notation 1/s indicates the process of time integration. The function INT (x) is defined as the value of x truncated to the greatest integer less than or equal to x. Whether quantization is achieved through *truncation or rounding* is irrelevant to the rate tracking problem. The model employed here will assume that truncation is employed. The superscript "*" indicates a quantized measurement. The caret ("^") is used to indicate an estimated quantity. The sampling rate is fixed at one sample every τ seconds. The nominal value of τ for BCAS is 1 second.

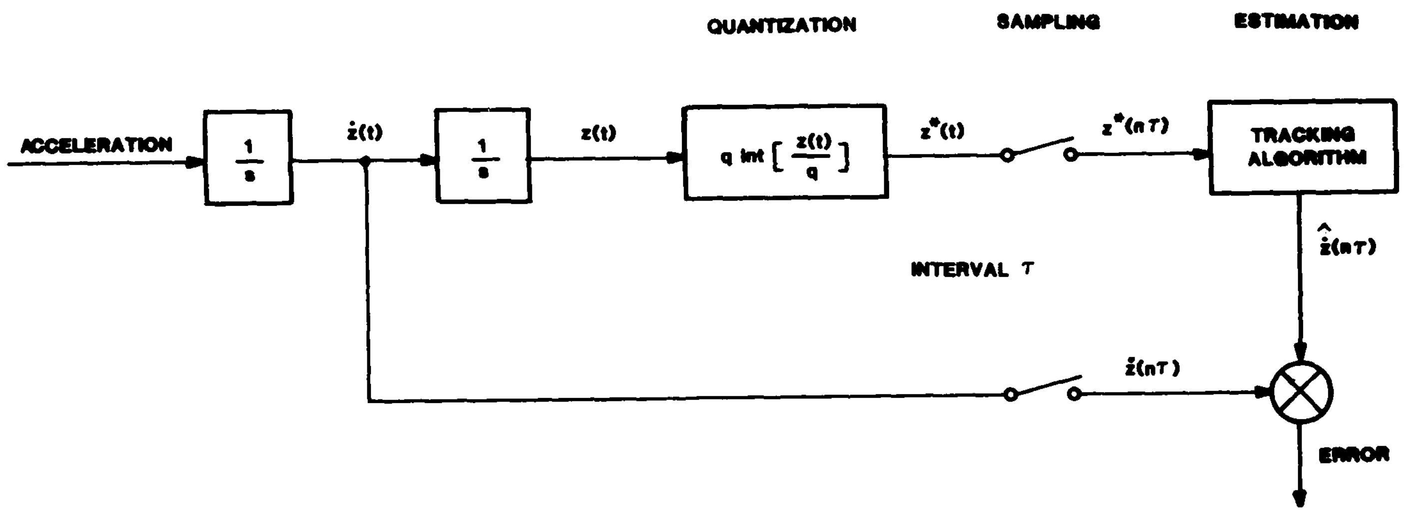

Fig. 2.1. System model of the altitude tracking process.

3.0 CHARACTERISTICS OF QUANTIZED SAMPLES

Certain characteristics of quantized samples will now be derived as a prelude to the consideration of estimation techniques. Mathematical relationships which are useful in the following derivations are provided in Appendix A of this document. Let the altitude of an aircraft be given by

$$z(t) = Nq + \varepsilon_0\, q + \dot{z}\, t \qquad (3.1)$$

where N is an integer and $0 \leq \varepsilon_0 < 1$. The quantity ε_0 defines the initial position of the aircraft within the quantization level of width q.

For simplicity, the following derivation will assume that $\dot{z} > 0$. The derivations retain their generality since the case of $\dot{z} < 0$ is obtained by merely reflecting the equations about the origin of the coordinate system. The quantized altitude at the n^{th} sample instant can now be written:

$$z^*(n\tau) = Nq + q\ \mathrm{INT}\left(\varepsilon_0 + \frac{n\dot{z}\tau}{q}\right)$$

$$= Nq + q\ \mathrm{INT}\left(\varepsilon_0 + \frac{n\tau}{T}\right) \qquad (3.2)$$

where T is the level occupancy time, i.e., the time required for z(t) to change by an amount q. It can be seen that the k^{th} altitude transition occurs on sample n_k which satisfies the following inequality:

$$\varepsilon_0 + n_k \frac{\tau}{T} \geq k > \varepsilon_0 + (n_k - 1)\frac{\tau}{T} \qquad (3.3)$$

Solving for n_k yields:

$$n_k = \frac{T}{\tau}(k - \varepsilon_0) + R\left[1 - R\left[\frac{T}{\tau}(k - \varepsilon_0)\right]\right] \qquad (3.4)$$

where R is defined as the function which yields the fractional remainder of the argument, i.e.,

$$R(x) = x - \mathrm{INT}(x)$$

The observed time within a single quantization level, T_k^*, is related to the n_k as follows:

$$T_k^* = \tau(n_{k+1} - n_k) \qquad k = 1,2,3,4,... \tag{3.5}$$

Note that at least two transitions must occur to produce the first observation of level occupancy. Substituting from equation 3.4 yields the fact that

$$T_k^* = \begin{cases} \tau \text{ INT } (T/\tau) & \text{for } R\left[(k-\varepsilon_0)\dfrac{T}{\tau}\right] < 1 - R(T/\tau) \\ \tau \text{ INT } (T/\tau) + \tau & \text{for } R\left[(k-\varepsilon_0)\dfrac{T}{\tau}\right] > 1 - R(T/\tau) \end{cases} \tag{3.6}$$

If T is an integer multiple of τ, then the observed level occupancy time will always be equal to T. Otherwise there are two possible values of T_k^*. These two values differ by τ and bracket the value of T. For randomly selected initial conditions the shorter value occurs with probability $1-R(T/\tau)$. The longer value occurs with probability $R(T/\tau)$. The value of T_k^* does not vary by more than τ for constant rate trajectories. This fact can be used in the detection of accelerations (as will be demonstrated later).

4.0 LINEAR RECURSIVE TRACKING

The original BCAS tracking algorithm uses the alpha-beta smoothing smoothing equations which follow:

$$\hat{z}_{n+1,n} = \hat{z}_n + \tau \hat{\dot{z}}_n \tag{4.1}$$

$$\hat{z}_{n+1} = \hat{z}_{n+1,n} + \alpha\,(z^*_{n+1} - \hat{z}_{n+1,n}) \tag{4.2}$$

$$\hat{\dot{z}}_{n+1} = \hat{\dot{z}}_n + \beta\,\frac{(z^*_{n+1} - \hat{z}_{n+1,n})}{\tau} \tag{4.3}$$

where $\hat{z}_{n+1,n}$ indicates the value of $\hat{z}_{n+1}$ as projected from the time of measurement z_n^*. Once a track is firmly established, the values of α and β are constant and these equations comprise a linear recursive tracking technique.

4.1 Step Response of the Alpha-beta Tracker

The response of the alpha-beta tracker to a change in altitude input of one quantization level will now be discussed. Consider a case in which an aircraft with negligible altitude rate crosses a quantization boundary. Assuming that the tracker estimate had previously converged to level flight at the reported altitude, the tracker is presented with a sudden discrepancy of magnitude q between the predicted and measured position. The characteristic response is sketched in Figure 4.1. The resulting rate estimates can be computed in closed form as shown in Table 4.1. For typical α and β values, the maximum rate error occurs on the second or third sample following the transition.

The maximum rate error for 1 second update interval is plotted in Figure 4.2 for a range of α and β values. Normally choices of α and β would be matched according to the formula

$$\beta = \frac{\alpha^2}{2 - \alpha} \qquad (4.4)$$

as suggested by Benedict and Bordner*. Points corresponding to this formula are indicated in the figure.

4.2 Tracking Cycle Behavior

For aircraft climbing at low to moderate rates, several samples are obtained at each quantization level. The rate estimates of the recursive alpha-beta tracker then tend to follow a tracking cycle in which the velocity is overestimated at the scan or scans immediately following a transition and underestimated for later scans. Depending on the value of the smoothing constants and the rate, the estimate near the end of the tracking cycle may converge to zero or even be opposite in sign to the direction of the transitions. This behavior is shown in Figure 4.3 for a climb rate of 800 FPM, β = 0.1.

---

*Benedict, T.R. and Bordner, G.W., "Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations," IRE Transactions on Automatic Control (July 1962).

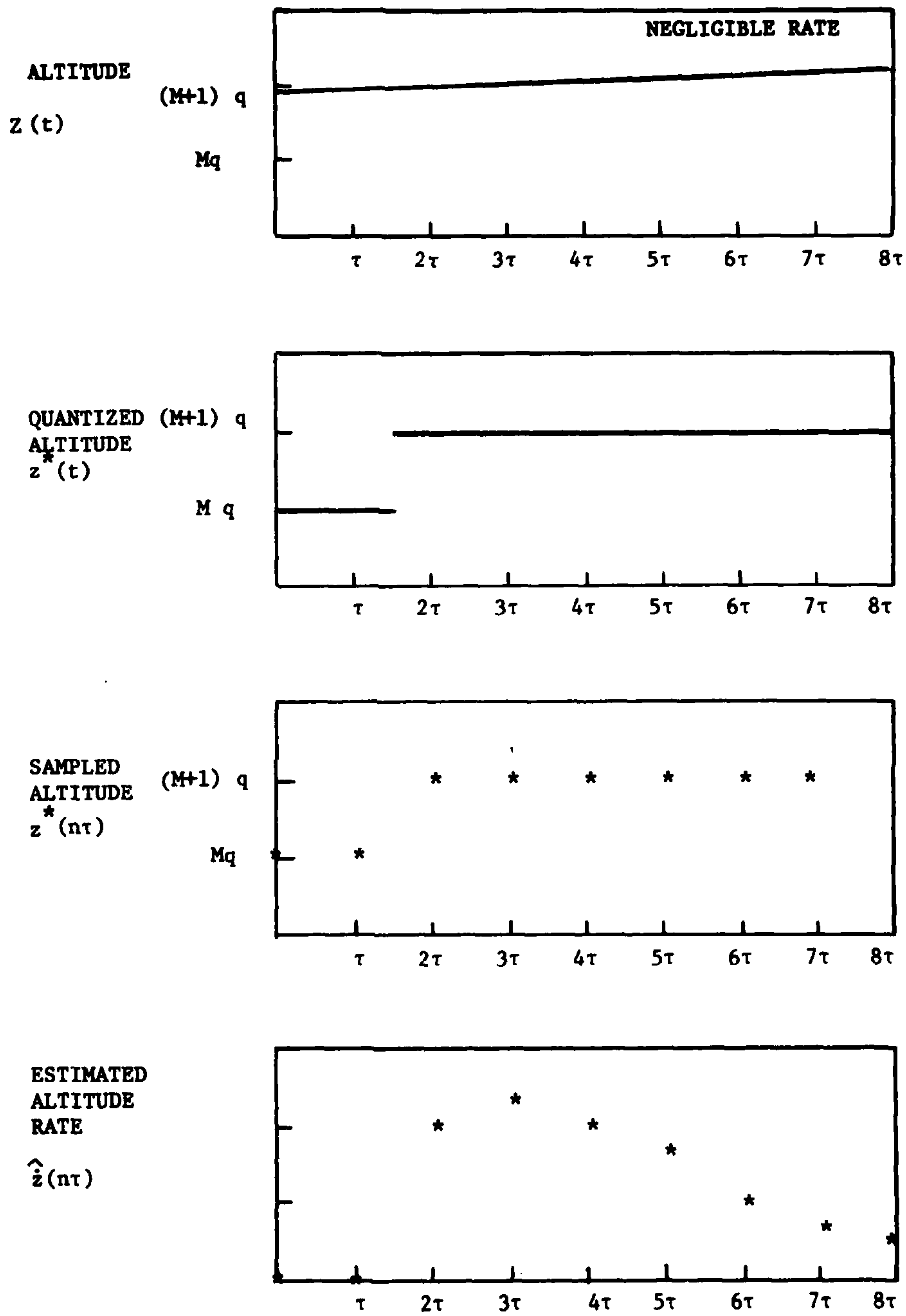

Fig. 4.1. Response of the α - β tracker to an isolated transition in quantized altitude.

TABLE 4.1

RESPONSE OF THE ALPHA-BETA TRACKER TO A SINGLE ALTITUDE TRANSITION

Scan (Altitude transition on scan N)	Position Response After Update	Rate Response After Update
N	αq	$\frac{\beta q}{\tau}$
N + 1	$(2\alpha - \alpha^2 - \alpha\beta + \beta)q$	$\frac{\beta q}{\tau}(2 - \alpha - \beta)$
N + 2	$(3\alpha - 3\alpha^2 - 5\alpha\beta + \alpha^3 - \beta^2 + 3\beta + 2\alpha^2\beta + \alpha\beta^2)q$	$\frac{\beta q}{\tau}(3 - 3\alpha - 4\beta + 2\alpha\beta + \alpha^2 + \beta^2)$

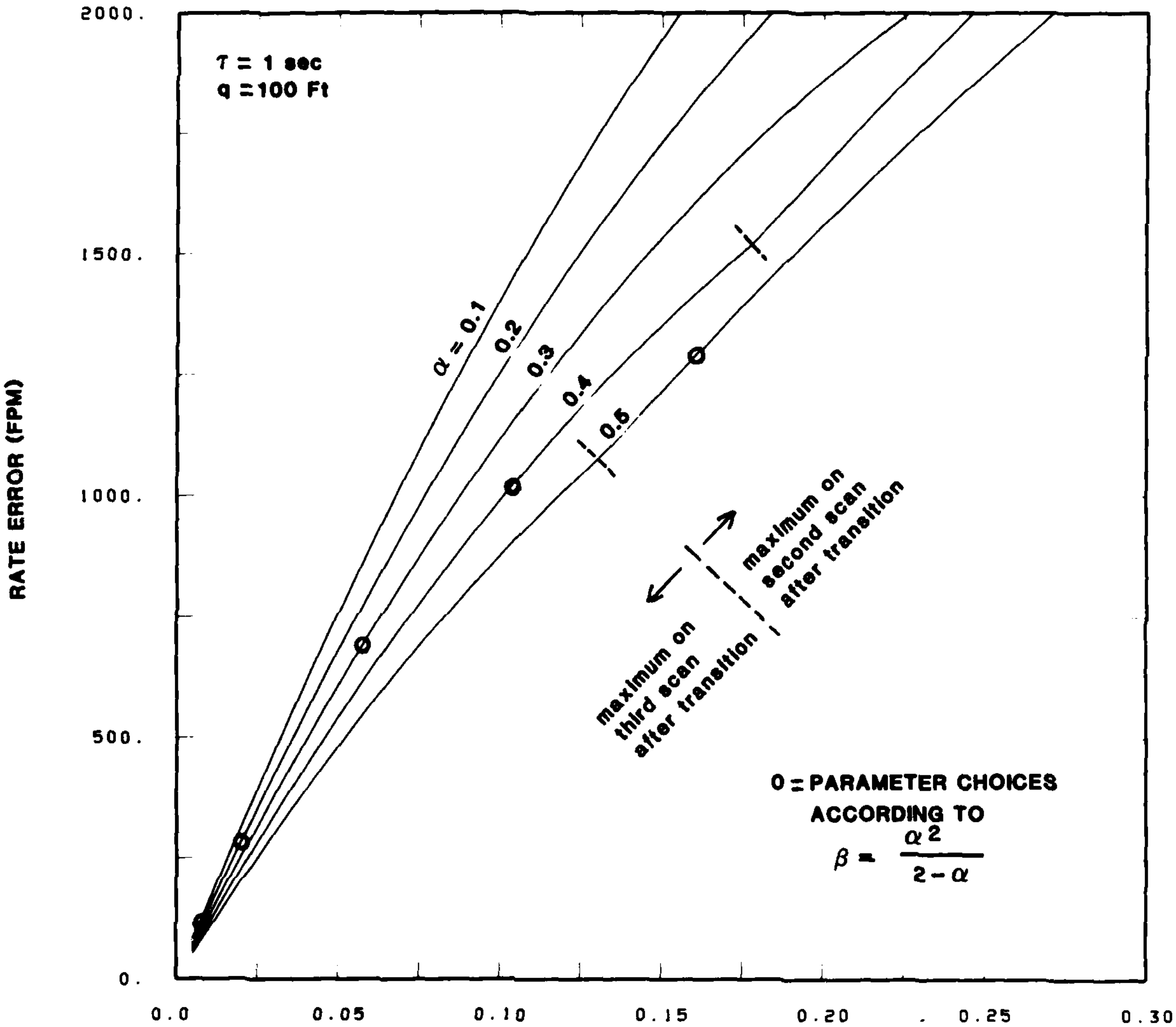

Fig. 4.2. Maximum rate error of an alpha-beta tracker due to a single 100 foot quantization level transition occuring in near-level flight. The errors shown are proportional to q/τ and hence the figure can be adapted to any update rate or quantization fineness by simply re-labeling the ordinate.

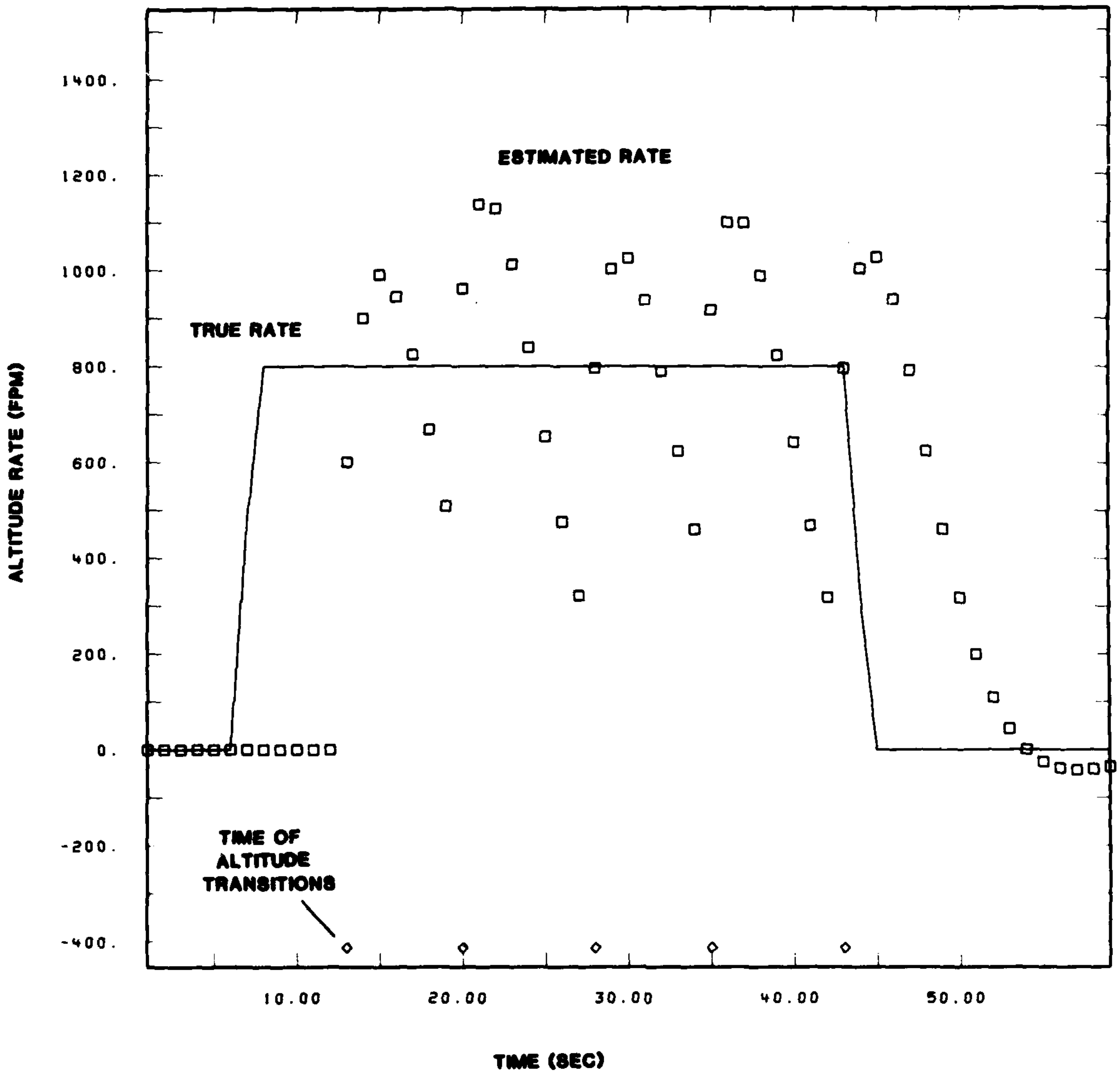

Fig. 4.3. Rate estimates of an α - β tracker for 800 FPM ramp (α = 0.4 β = 0.1).

5.0 LEVEL OCCUPANCY TRACKING

5.1 Motivation for Level Occupancy Tracking

The underlying reason for the undesirable tracker behaviors described above is that the error characteristics of the measurement system diverge from those characteristics for which the alpha-beta tracker is best suited. The alpha-beta tracking treats each data point as an independent measurement (as if errors were uncorrelated from sample to sample). In reality, the errors can be highly correlated, especially when the rate is such that multiple samples are obtained in the process of crossing a single quantization level. A little thought reveals that all the rate information is contained in the history of altitude transitions. Increasing the sampling rate benefits tracking accuracy not so much by providing more data points at each level as by decreasing the error in the determination of the level transition times.

The above observations suggest a change in the approach to altitude tracking. In order to avoid the problems of the recursive update at each sample point, all data points which fall within the same altitude quantization bin are treated as a single observation of rate. The smoothing equations are then written in terms of level occupancy time, T, the amount of time required for the aircraft to cross a single quantization level. The true value of this time is

$$T = \frac{q}{|\dot{z}|} \tag{5.1}$$

The result of the measurement process is then viewed as a series of observations of past occupancy times plus an observation of the time at the current level.

The difference between any two successive altitude transitions serves as a measurement of T. Errors in this measurement are attributable to the finite sampling rate. The effect of these errors on the rate estimate are reduced by smoothing successive values of the level occupancy time.

Note that in this formulation of the problem, an altitude measurement which yields the same altitude value as the previous measurement does not provide a new measurement of T and hence is not smoothed. The tracking cycle behavior is thus eliminated. However the lack of a transition may be significant if the aircraft has occupied the current level for longer than expected - in that case the lack of a transition may indicate that an acceleration has occurred which requires that the magnitude of the rate estimate be decreased. Hence an altitude measurement without transition requires a special check to determine if a transition is overdue. The differing update procedures are indicated schematically in Figure 5.1.

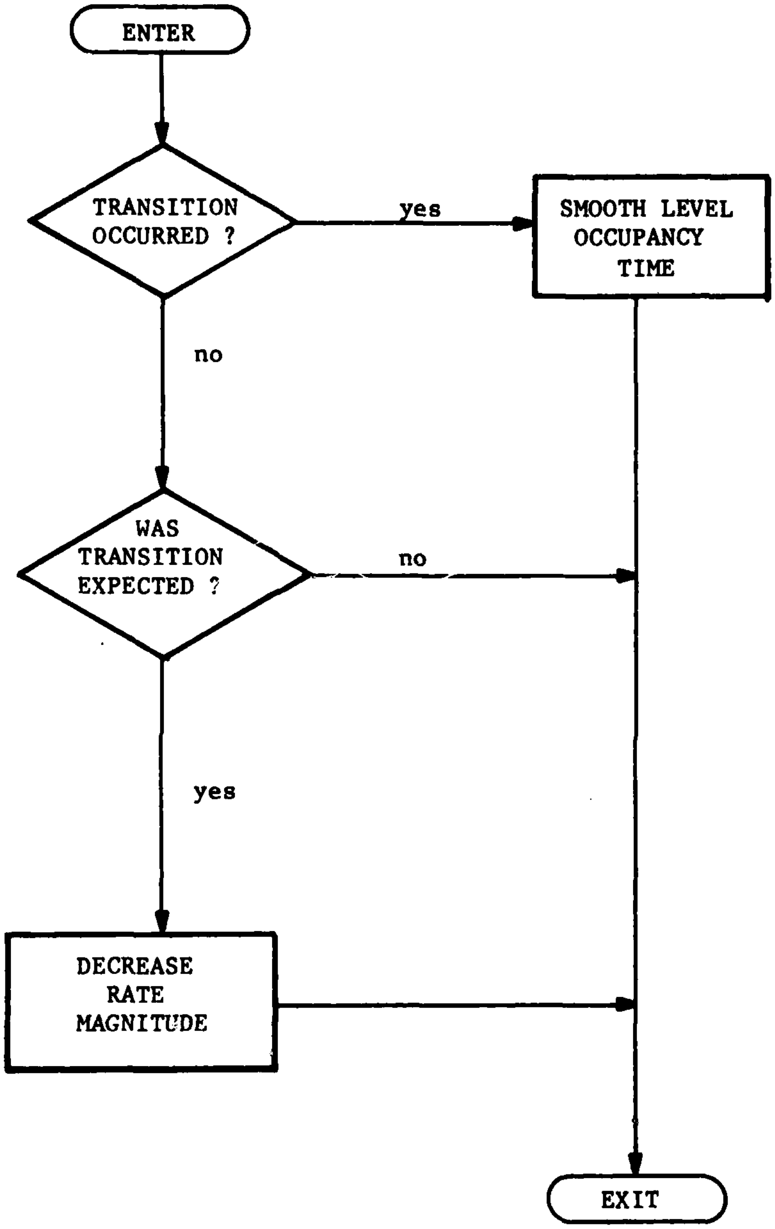

Fig. 5.1. Basic update procedure for level occupancy tracking.

5.2 Analysis of Rate Tracking Accuracy

The effect of quantization upon tracking accuracy will now be examined for an algorithm which functions by smoothing level occupancy time. The observed value of the level occupancy time is the time difference between observed altitude transitions. Hence the observed occupancy time is always a multiple of the fundamental sample rate τ.

Let the first observed transition occur at time t_0 when the aircraft is a distance $\varepsilon_0 q$ from the level boundary most recently crossed. This situation is shown in Figure 5.2. Note that in this notation $0 \leq \varepsilon_0 < \tau/T < 1$. Since the direction of the transition determines the sign of the rate, the sign can be viewed as a known quantity. Hence we may, without loss of generality, discuss only the case for $\dot{z} > 0$. The position change between the first and second transitions of the sampled data is

$$\dot{z}\, T_1^* = q + \varepsilon_1 q - \varepsilon_0 q \qquad (5.2)$$

and hence

$$T_1^* = T\,(1 + \varepsilon_1 - \varepsilon_0) \qquad (5.3)$$

Furthermore it is obvious that in the general case

$$T_j^* = T\,(1 + \varepsilon_j - \varepsilon_{j-1}) \qquad (5.4)$$

Consider the effect of averaging the level occupancy time over k observations. The estimate of T which results can be written as

$$\hat{T}_k = \frac{\sum_{j=1}^{k} T_j^*}{k} = T + T\,\frac{(\varepsilon_k - \varepsilon_0)}{k} \qquad (5.5)$$

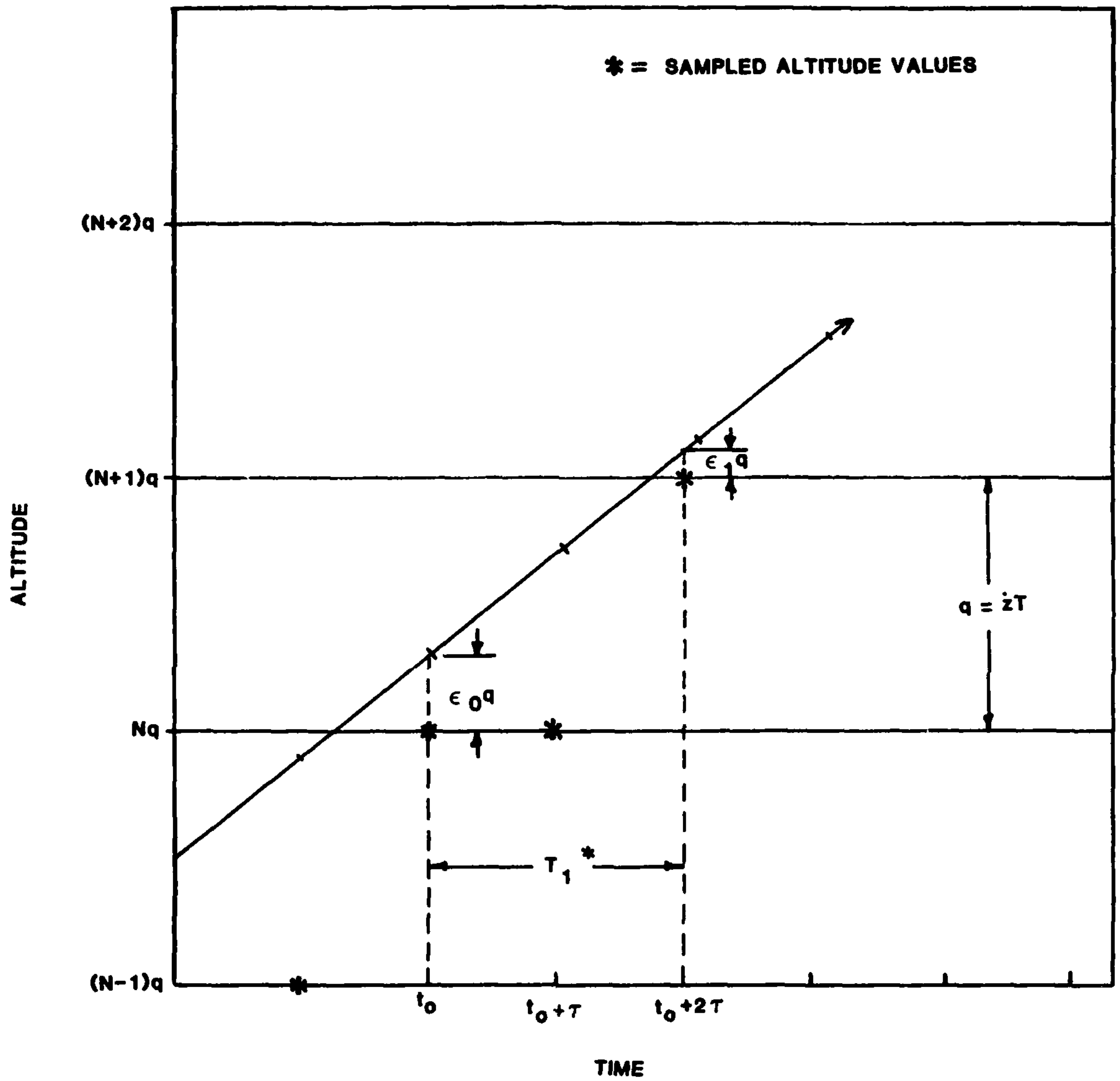

Fig. 5.2. Notation employed to describe a constant rate altitude profile.

The error in $\hat{T}$ is

$$e_{\hat{T}} = T \frac{\varepsilon_k - \varepsilon_0}{k} \tag{5.6}$$

This error is proportional to the difference in the initial and final values of the quantization residual ε and is inversely proportional to the number of level occupancy transitions observed.

The value of ε can be no more than τ/T on the scan at which a transition is observed. Thus $0 \leq \varepsilon_j < \tau/T$, $j = 0,1,2,\ldots$, and the magnitude of the error is limited according to

$$|e_{\hat{T}}| \leq \frac{\tau}{k} \tag{5.7}$$

Under the assumption that ε_k and ε_0 are independent and uniformly distributed over $[0,\tau/T]$, the quantity $\varepsilon = \varepsilon_k - \varepsilon_0$ has the probability density function

$$f_\varepsilon(x) = \begin{cases} \left(\frac{T}{\tau}\right)^2 \left(\frac{\tau}{T} - |x|\right) & -\frac{\tau}{T} \leq x \leq \frac{\tau}{T} \\ 0 & \text{Otherwise} \end{cases} \tag{5.8}$$

Using 5.8 it can readily be shown that $E(e_{\hat{T}}) = 0$, i.e., the estimate of T is unbiased. The varience in $\hat{T}$ can be shown to be:

$$\sigma^2_{\hat{T}^2} = \frac{\tau^2}{6\,k^2} \tag{5.9}$$

and hence

$$\sigma_{\hat{T}} = \frac{\tau}{k\sqrt{6}} \tag{5.10}$$

This error, expressed as a fraction of τ, is inversely proportional to the number of observed level occupancy intervals. This expression is plotted in Figure 5.3. It should be noted that no estimation of T is possible until at least two altitude transitions have been observed. The error is never greater than the sample interval τ.

It should be noted that the error converges faster than would be expected for a case where measurement errors were normally distributed white noise. In the latter case, $\sigma_{\hat{T}}$ decreases according to $1/\sqrt{k}$ rather than $1/k$.

If the rate estimate is simply $\hat{\dot{z}} = q/\hat{T}$, then the error in $\hat{\dot{z}}$ at the k^{th} observed occupancy time can be expressed as a fraction of $\dot{z}$ as follows:

$$e_{\hat{\dot{z}}} = \frac{\hat{\dot{z}}(t_k) - \dot{z}}{\dot{z}} = \frac{\varepsilon_0 - \varepsilon_k}{k + \varepsilon_k - \varepsilon_0} \tag{5.11}$$

The value of $\varepsilon_k - \varepsilon_0$ satisfies the following inequality

$$-\frac{\tau}{T} < \varepsilon_k - \varepsilon_0 < \frac{\tau}{T} \tag{5.12}$$

Hence the maximum value of $e_{\hat{\dot{z}}}$ is:

$$e_{\hat{\dot{z}}} = \frac{\tau/T}{k - \tau/T} = \frac{1}{\frac{kT}{\tau} - 1} \tag{5.13}$$

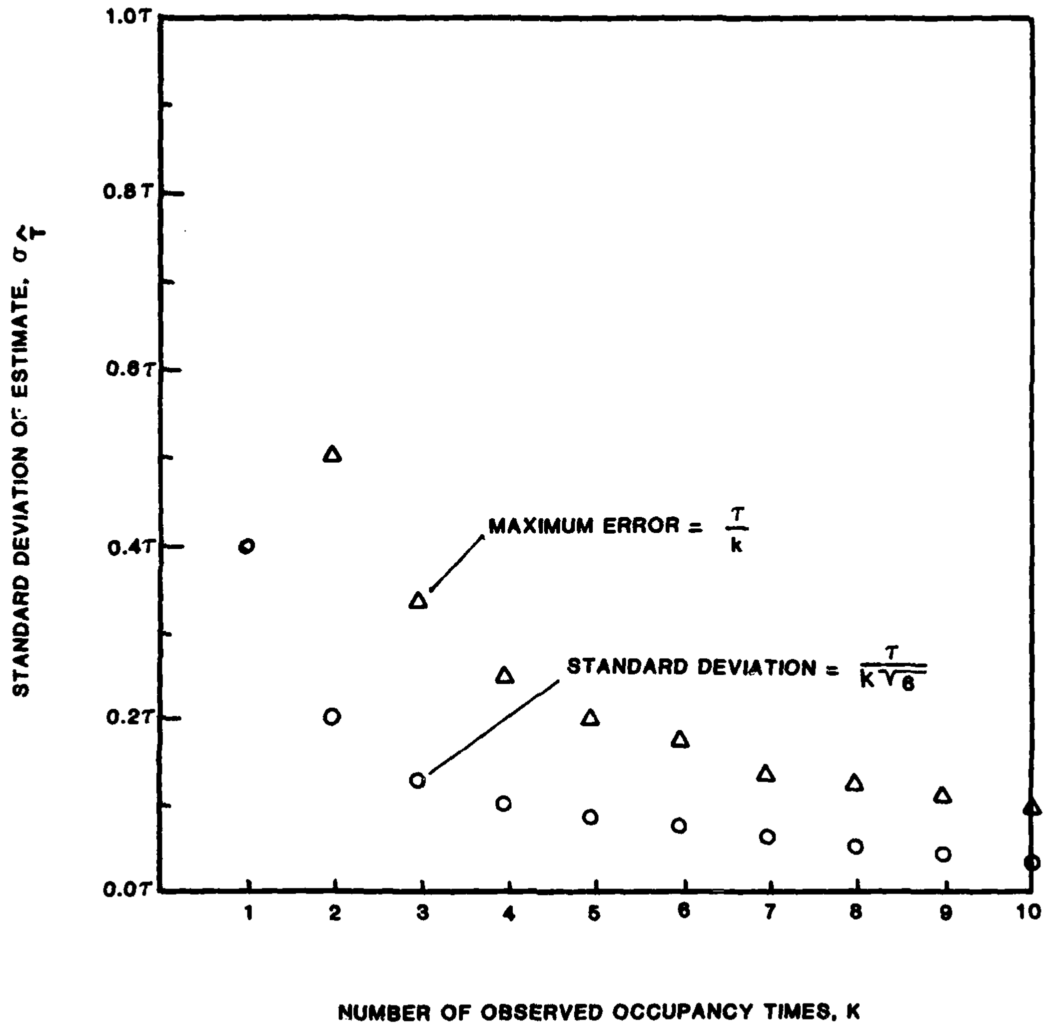

Fig. 5.3. Summary of errors in the estimation of T after K observed level occupancy times. Note that the number of observed level transitions is K+1.

5.3 Position Correction

As previously discussed in Section 2.2, it is generally not possible to determine exactly where an aircraft is within a quantization level. However, it can be seen from Figure 5.2 that on the scan upon which a transition occurs, the aircraft must be within a distance of $q\tau/T$ of the quantization boundary most recently crossed. Thus, if the boundary crossed is at Nq, (N = 1,2,3 ...), then the altitude is between Nq and $Nq + \dot{z}\tau$. A tracking algorithm can use this fact by always correcting the position estimate so that it lies within this interval. In the algorithm described later, the midpoint of the indicated altitude interval is selected as the new estimated position, i.e.,

$$\hat{z} = Nq + 1/2\, \dot{z}\, \tau$$

This correction is of greatest benefit when $T >> \tau$ and is of negligible benefit when $T \cong \tau$.

6.0 ALGORITHMIC PROVISIONS FOR TRACKING RATE CHANGES

The preceeding section introduced the concept of rate estimation through smoothing of level occupancy times. The results obtained by simple averaging of observed occupancy times were analyzed. Although this averaging technique can form the basis for estimation, an algorithm is not complete without provisions for adequate tracking in the case of non-constant rates. Three such provisions are discussed in this section. The first is a provision for decreasing the magnitude of the estimated rate when expected level transitions fail to occur. The second is a technique for preventing the gain of the tracker from converging to zero during long periods of consistent transitions. Finally, a procedure for testing the consistency of the estimated and observed occupancy times is described. A tracking algorithm which combines all these techniques is provided in Appendix B.

6.1 Update Procedure for Decreases in Rate Magnitude

As described in Section 5.1, the algorithm under consideration smooths the level occupancy time only when altitude transitions occur. A distinct update procedure must be invoked whenever expected transitions <u>fail</u> to occur. The procedure for detection of overly long level occupancies is based upon the following test for consistency between observed and estimated level occupancy times.

Let S_j be the difference between the estimated and the observed value of T on the jth transition. That is

$$s_j = \hat{T}_{j-1} - T_j^* \qquad j = 2,3,4.... \qquad (6.1)$$

It has been shown in equation 3.6 that the observed value of T always falls between $\tau INT(T/\tau)$ and $\tau INT(T/\tau) + \tau$. Since $\hat{T}_{j-1}$ is merely an average of previous observed level occupancy times it is within the same range of width τ. Thus the residual s_j has magnitude of τ or less:

$$|s_j| < \tau \tag{6.2}$$

Barring acceleration, an observed occupancy time should never differ from the previous estimate by more than τ. Hence if the current time of level occupancy (i.e., the time since the last transition) is equal to or greater than $\hat{T} + \tau$, then an acceleration which has decreased the rate magnitude is indicated. The proper response of the filter to this situation depends upon the assumed aircraft trajectory statistics. A minimal response would be to decrease the rate estimate only as much as required to make it consistent with the observed occupancy time. But it has been found that this results in a very slow convergence to zero which is undesirable when the aircraft has leveled out at the end of a climb or descent.

Under the assumption that a return to level flight is more likely than a transition to another non-level altitude rate, a better approach is to force a rather fast convergence to zero when the data clearly indicates that an acceleration toward zero rate has occurred. This is the approach taken in the algorithm described in Appendix B. A variable which indicates excess occupancy time is defined by

$$\delta = \frac{T^* - \hat{T}}{\tau} \tag{6.3}$$

where T^* is observed time in the current quantization level and $\hat{T}$ is the current estimate of the level occupancy time. Whenever δ is greater than 1, the rate is adjusted toward zero according to an experimentally derived formula. When δ is greater than 5, it is assumed that the aircraft has returned to level flight.

6.2 Lower Limit on Tracker Gain

The process of averaging level occupancy times can be carried out in recursive fashion by use of the following smoothing equation:

$$\hat{T}_n = \hat{T}_{n-1} + \beta_n (T_n^* - \hat{T}_{n-1}) \tag{6.4}$$

where

$$\beta_n = 1/n \tag{6.5}$$

and

$$\hat{T}_0 = 0 \tag{6.6}$$

Note that as the number of smoothing updates increases, the gain of the tracker (as represented by β_n) decreases toward zero. This is appropriate only if the rate is constant. In actual operation, the rate is never exactly constant and hence the convergence of β_n to zero must be slowed or interrupted.

A classic least squared error approach to tracking in the presence of state perturbations results in a tracker gain which decreases toward a limit which is determined by the ratio of the measurement error to the magnitude of the perturbation. In the current case the perturbation is related to the amount of velocity change which is expected between smoothing instants. These instants occur at time intervals of approximately T. There is a greater expected perturbation when T is large. This implies that the lower limit on β_n should be smaller for smaller T, larger for larger T. (Another way of looking at this is that in order to implement a "fading memory" tracker, one must assign less weight to observations the further they are in the past. The smaller the value of T, the more recent were past observations, and the more heavily they should be weighted relative to the current observation). Although the true value of T is not available to the tracker, it is sufficient to select the limit on the basis of T. In the tracking algorithm tested in simulation, a β limit was selected according to the following expression:

$$\beta_{min} = \max \left[\frac{(\hat{T} - 1)^2}{\hat{T}^2 + 64} , 0.08 \right] \qquad (6.7)$$

This expression was developed by first selecting a form which satisfied reasonable "end point" criteria. Parameter values were then optimized through simulation. The limit it imposes is plotted in Figure 6.1.

6.3 Consistency Tests Applied to Level Transitions

This section discusses criteria used to determine whether the time of an observed level transition is consistent with the existing rate estimate. Inconsistencies indicate that acceleration has occurred and that the existing estimate may have substantial error. The response to this situation is either to reinitialize the track or to increase the tracker gain.

6.3.1 Sign Consistency

The direction of the altitude transition is always the same as the direction of the true rate. Note that sign consistency is not guaranteed in the simple alpha-beta tracker and may result 1) when rate reversals occur or 2) as a consequence of the tracking cycle behavior discussed in 4.2 as the rate estimate oscillates around zero. In the algorithm formulated in Appendix B, sign consistency is required at all times. This means that whenever the sign of the transition is opposite to the sign of the rate estimate, the track is reinitialized.

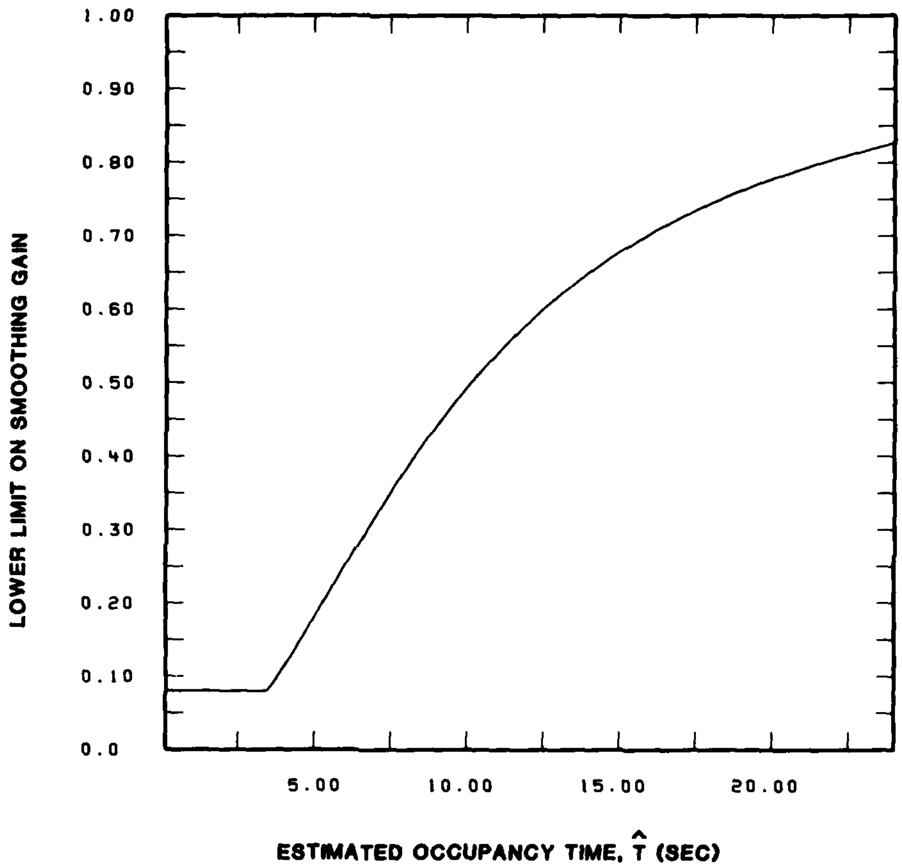

Fig. 6.1. Lower limit on smoothing gain.

6.3.2 Single-scan Consistency Test

As shown in Section 6.1, the magnitude of the residual

$$s_j = \hat{T}_{j-1} - T_j^* \tag{6.8}$$

should never exceed τ. Hence if $|s_j|$ is greater than τ, acceleration has occurred and reinitialization is desirable. When such reinitialization occurs, averaging begins anew and smoothing parameter β_n of Equation 6.4 should be reset to the value for $n=1$.

6.3.3 Test of Summed Residuals

If an acceleration alters the value of T by an amount less than τ, the residual may never exceed τ (or may not exceed τ until several level occupancy intervals have passed). The single scan consistency test will not result in detection. But acceleration may still be detectable by a test which sums residuals over more than one transition. This test is based upon the fact that the sum of the residuals tends toward zero except when $\hat{T}$ is in error. The test is implemented by computing, at each smoothing, a weighted sum of the residuals according to the formula

$$\bar{s}_j = \gamma \, \bar{s}_{j-1} + s_j \tag{6.9}$$

Whenever $\bar{s}_j$ exceeds a certain threshold, an excess residual is declared. The value of $\bar{s}_j$ is reset once such a detection has occurred. The parameter γ is set to slightly less than unity to provide a gradual reset of $\bar{s}_j$ in periods within which no detection occurs. In the algorithm described in Appendix B, the value of γ is 0.8 and the detection threshold for $\bar{s}_j$ is 1.35.

7.0 SIMULATION RESULTS FOR A LEVEL OCCUPANCY TRACKER

The performance benefits to be derived from the tracking techniques developed in the previous sections cannot be adequately evaluated without testing a complete algorithm which properly integrates the various tracking features. This section introduces one such integrated algorithm and presents simulation results which demonstrate its performance in a variety of situations.

7.1 Description of the Tracking Algorithm

A complete description of the algorithm used for simulation is provided in Appendix B. Several details of this algorithm (including the choice of parameter values) were derived by experimentation with a range of options. No attempt is made here to present the intermediate results which led to this final form. However, careful inspection of the simulation data supports the contention that the final algorithmic form is "near optimum" in that it achieves almost all the performance improvement which can be expected from level occupancy tracking.

The basic features of the algorithm can be summarized as follows:

a. A single level transition following an extended period of constant altitude flight results in initialization of the rate estimate magnitude to a nominal value of 480 FPM (note parameter P1). This rate decays by 10% (note parameter P3) on each successive scan without a transition.

b. The second transition, if consistent in sign with the first, results in initialization of $\hat{T}$ to the observed time between transitions. No routine decay is then allowed.

c. All subsequent transitions which satisfy consistency tests result in recursive averaging of the observed bin occupancy times.

d. If the time for which a level has been occupied exceeds by 1.5τ to 5τ (note parameters P5 and P6) the time predicted by $\hat{T}$, the rate is driven toward zero by an empirically determined formula. If the excess time is greater than 5τ, a return to constant altitude flight (zero rate) is effected.

e. If at any point an inconsistency is noted in the sign of the rate estimate and the direction of a transition, the rate estimate is re-initialized as in (a).

f. If at the time of a transition, the observed occupancy time differs from the estimate by more than τ, the rate is reinitialized to the rate corresponding to the observed occupancy time. This provides a quick response to a major change in an established rate.

g. If the smoothed residual of equation 6.9 exceeds τ at the time of any transition, the gain of the tracker is increased and the convergence of the gain towards its lower limit is set back.

7.2 Simulation Results at 1-Second Update Interval

The series of figures which follow compare the rate tracking performance of the level occupancy tracker with simple alpha-beta tracking. The simulation employed no measurement jitter and used a constant update interval of 1 second. All aircraft rate changes took place with an acceleration magnitude of 0.25g (8ft/sec^2).

7.2.1 Step Response

The response of the trackers to a single isolated altitude transition (a "step" in altitude) is shown in Figure 7.1. As previously noted, the step response of the level occupancy tracker is arbitrary, being determined by the parameters P1 and P3. In BCAS testing the response of the $\beta = 0.1$ tracker was found to be unacceptable in this situation. A value of $\beta = 0.05$ was later used for BCAS alpha-beta tracking and found to offer acceptable step function response. It can be seen that the level occupancy tracker appears to be of essentially equivalent acceptability.

7.2.2 Ramp Response

The response of the trackers to various simple climb profiles will now be shown. A "ramp" climb profile will be defined as a profile which involves an initial period of level flight, acceleration to a specified rate, a period of constant rate climb, acceleration back to zero rate, and a final period of level flight. Generally the largest tracking errors will occur during or immediately following the periods of acceleration. The magnitude of the errors is strongly dependent upon the ramp rate magnitude. Figure 7.2 plots rate estimates for a 450 FPM ramp. The tracking cycle behavior of the alpha-beta tracker is quite obvious here, even with the reduced tracker gain. Note that the level occupancy tracker eliminates the tracking cycle behavior after the second transition. When the climb terminates, the drifting of the alpha-beta tracker back toward zero results in a fortuitous convergence toward the true rate.

Figure 7.3 plots the ramp response for an 800 FPM ramp. Here the alpha-beta parameters are well matched to the rate and convergence for both trackers seems to occur at about the second transition. The tracking cycle behavior is noticeable, but is a smaller fraction of the true rate.

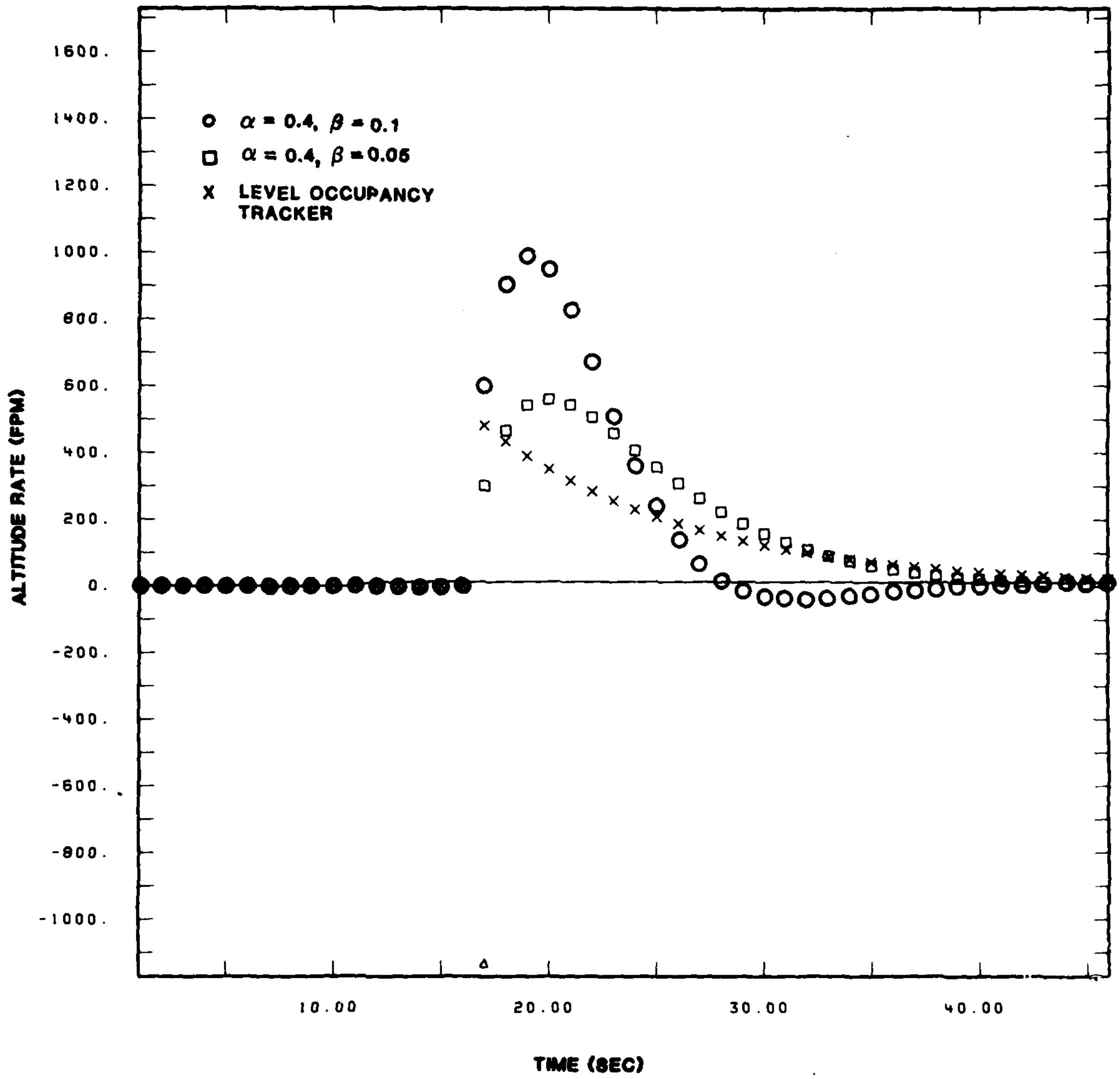

Fig. 7.1. Response of trackers to a single transition in quantized altitude.

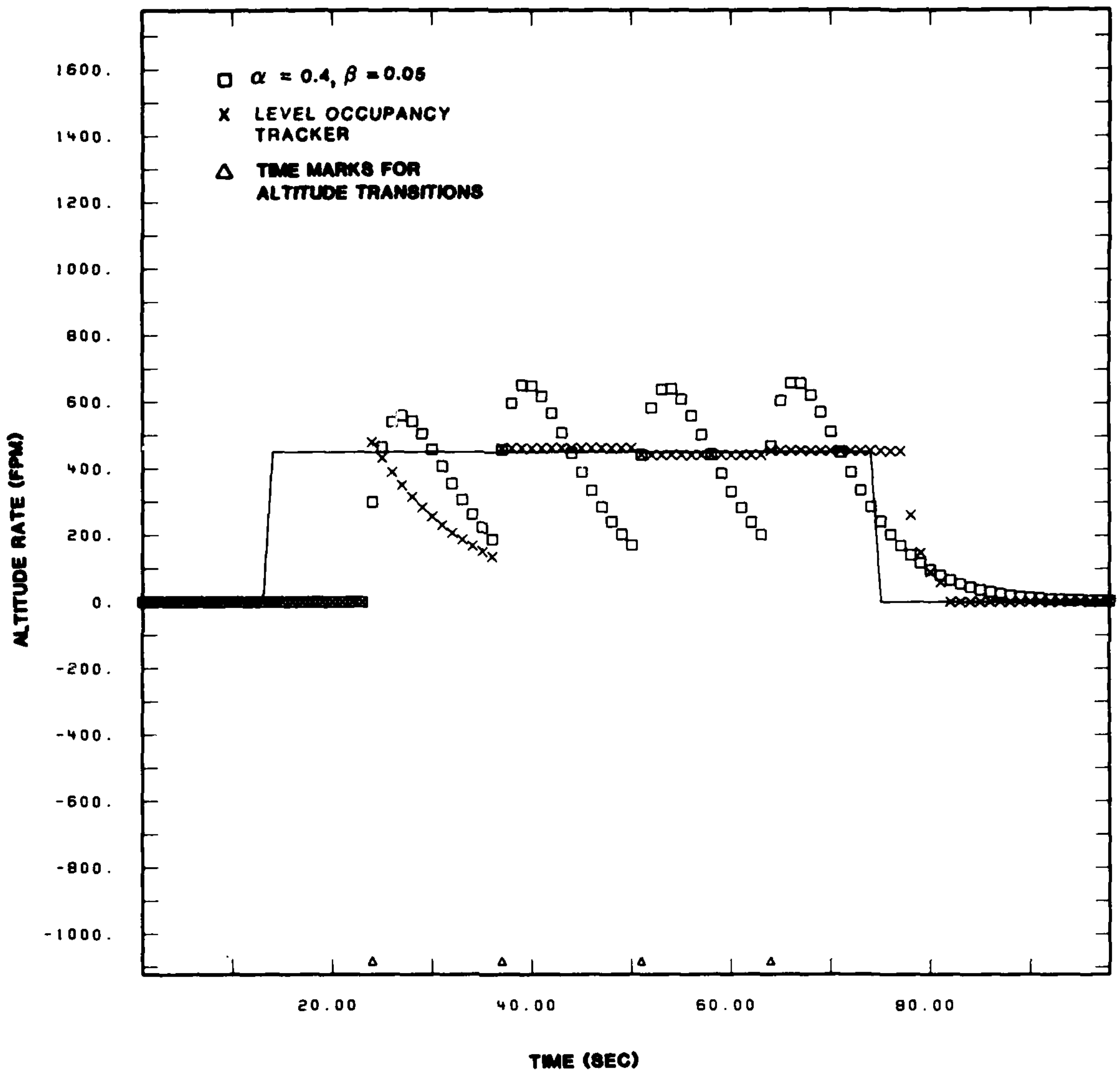

Fig. 7.2. Rate estimation performance for a 450 FPM ramp profile.

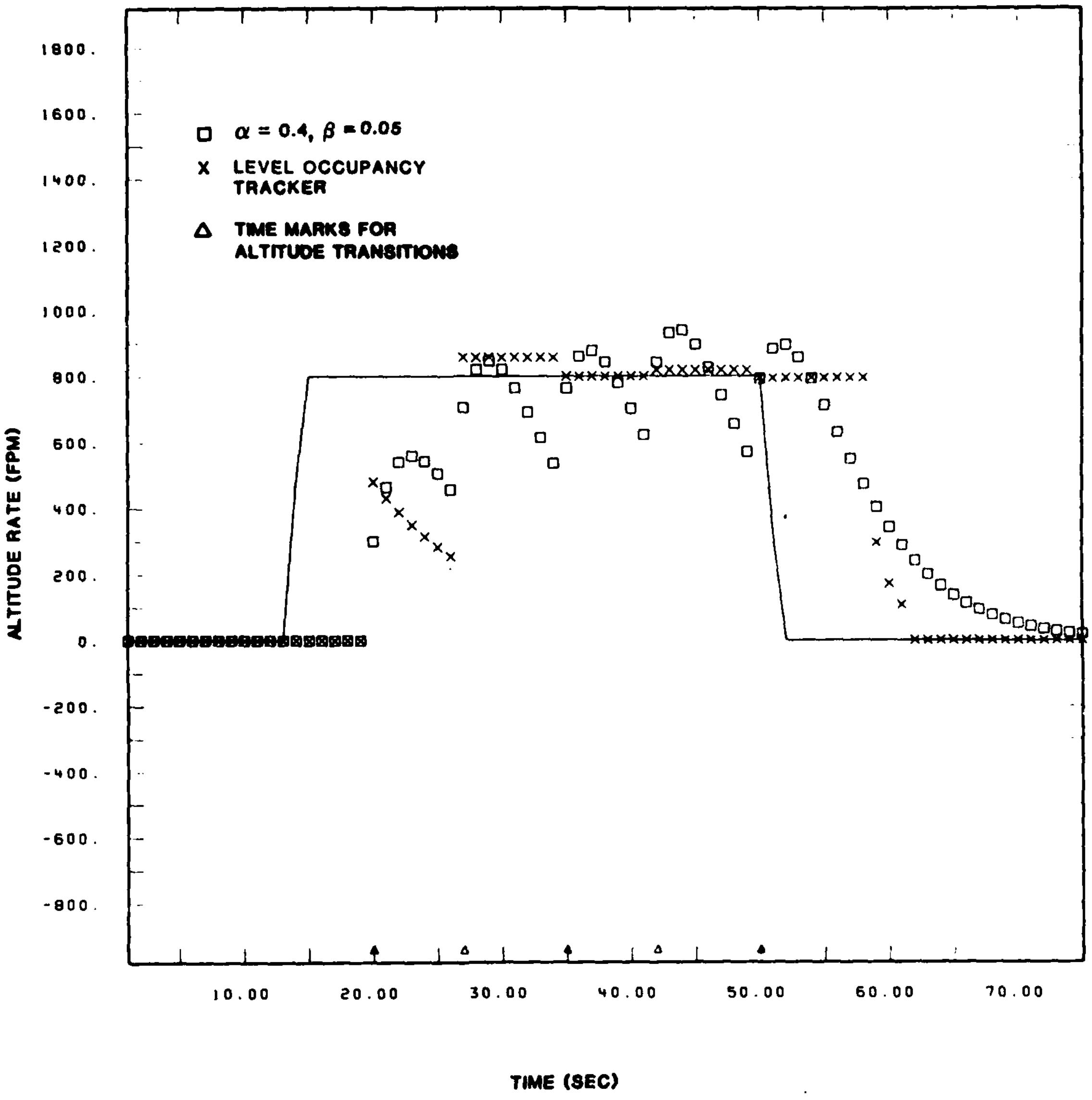

Fig. 7.3. Rate estimation performance for a 800 FPM ramp profile.

Figure 7.4 provides results for a 2100 FPM ramp. At this higher rate it can be seen that the alpha-beta tracker does not converge to the true rate until about the fifth altitude transition (13 seconds after the first transition). There is a period of time after the second transition when the rate is severely underestimated in comparison to the level occupancy tracker. Similarly, the alpha-beta tracker requires many scans to recover when the climb terminates.

Figure 7.5 presents results for a 5000 FPM ramp. For this high final rate the period of acceleration is prolonged and several transitions occur during the acceleration. The level occupancy tracker appears to do better than the alpha-beta tracker during and following the acceleration periods.

One meaningful measure of performance is the number of scans on which the magnitude of the rate error exceeds a given threshold of significance. This measure indicates the amount of time for which the system is vulnerable to failures caused by tracking error. Figure 7.6 summarizes the ramp performance of the alpha-beta and level occupancy trackers using such an error count. The number of scans on which the magnitude of the rate estimation error exceeded 600 FPM is given for a range of ramp rates. For rates below 1000 FPM, this measure shows little apparent differences in tracker performance. But as rates exceed 1500 FPM, the advantage of the level occupancy tracker becomes significant.

It should be noted that in most cases a change in reported altitude of only 200 feet (2 transitions) is required in order for the tracker to converge to a reasonable estimate of the vertical rate. Hence for the level occupancy tracker it is more appropriate to think in terms of the number of transitions required for convergence rather than the length of time required. The tracker converges very rapidly when the vertical rate is high, since high rates quickly produce the required number of altitude level transitions.

7.2.3 Steady State Performance

RMS rate errors for steady state climbs at various rates are shown in Figure 7.7. Although the steady state errors are never large for either tracker, the level occupancy tracker exhibits smaller errors, especially for lower rates (below 1500 FPM).

8.0 TRACKING WITH LESS FREQUENTLY SAMPLED DATA

Ground-based air traffic control sensors typically possess update intervals of more than 4 seconds. At these sampling rates, aircraft with rate magnitudes of more than 1500 feet per minute will cross more than one quantization level between samples. Several modifications must be made to the tracking algorithm described in Appendix B in order to apply it to rate tracking in this regime:

- The value of the algorithmic parameters must be adjusted.

- The smoothing equation must be generalized to accomodate transitions of more than one quantization level.

- The consistency tests must be extended to consider the number of <u>transitions between samples</u> as well as the number of <u>samples between transitions</u>.

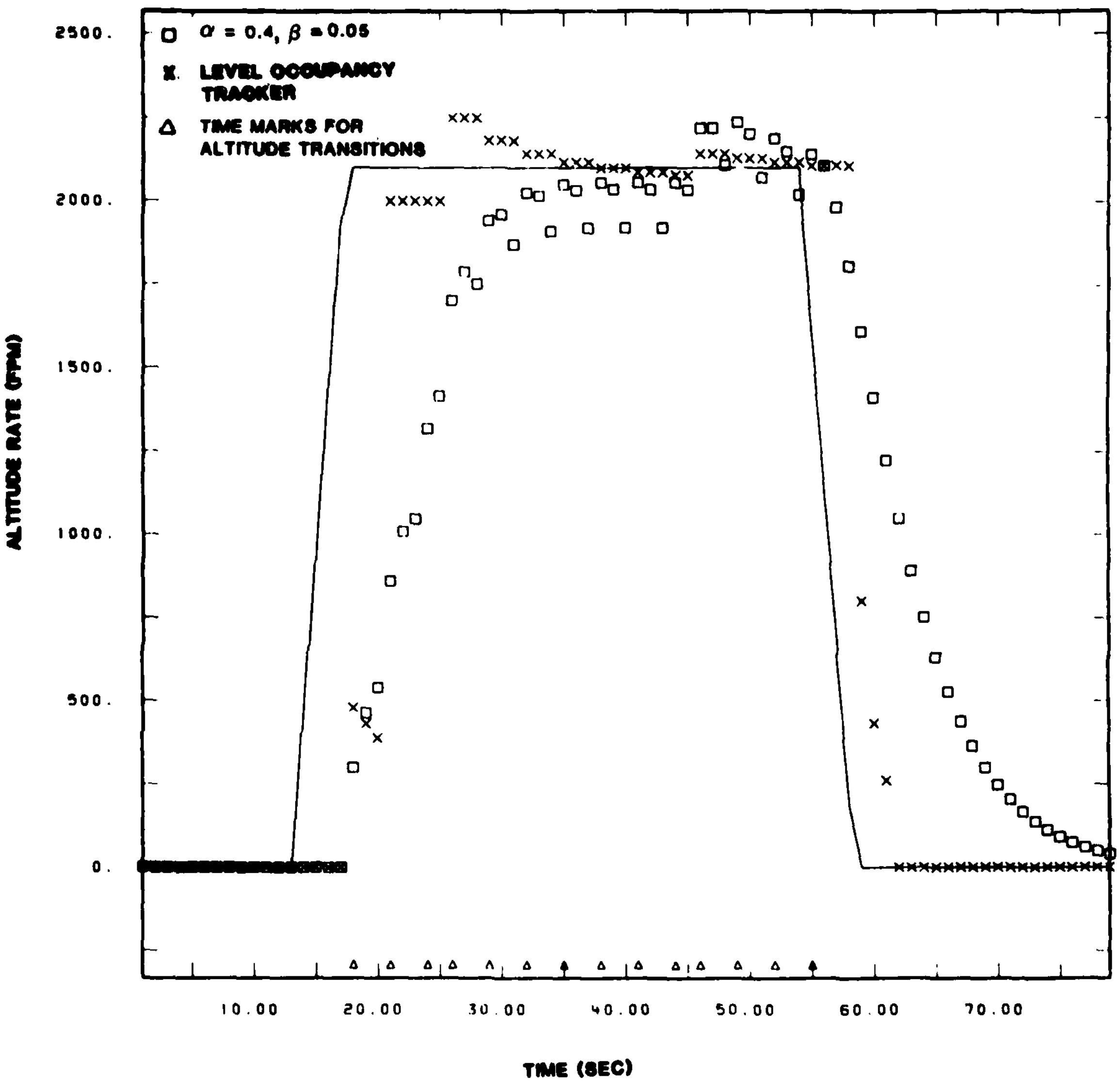

Fig. 7.4. Rate estimation performance for a 2100 FPM ramp profile.

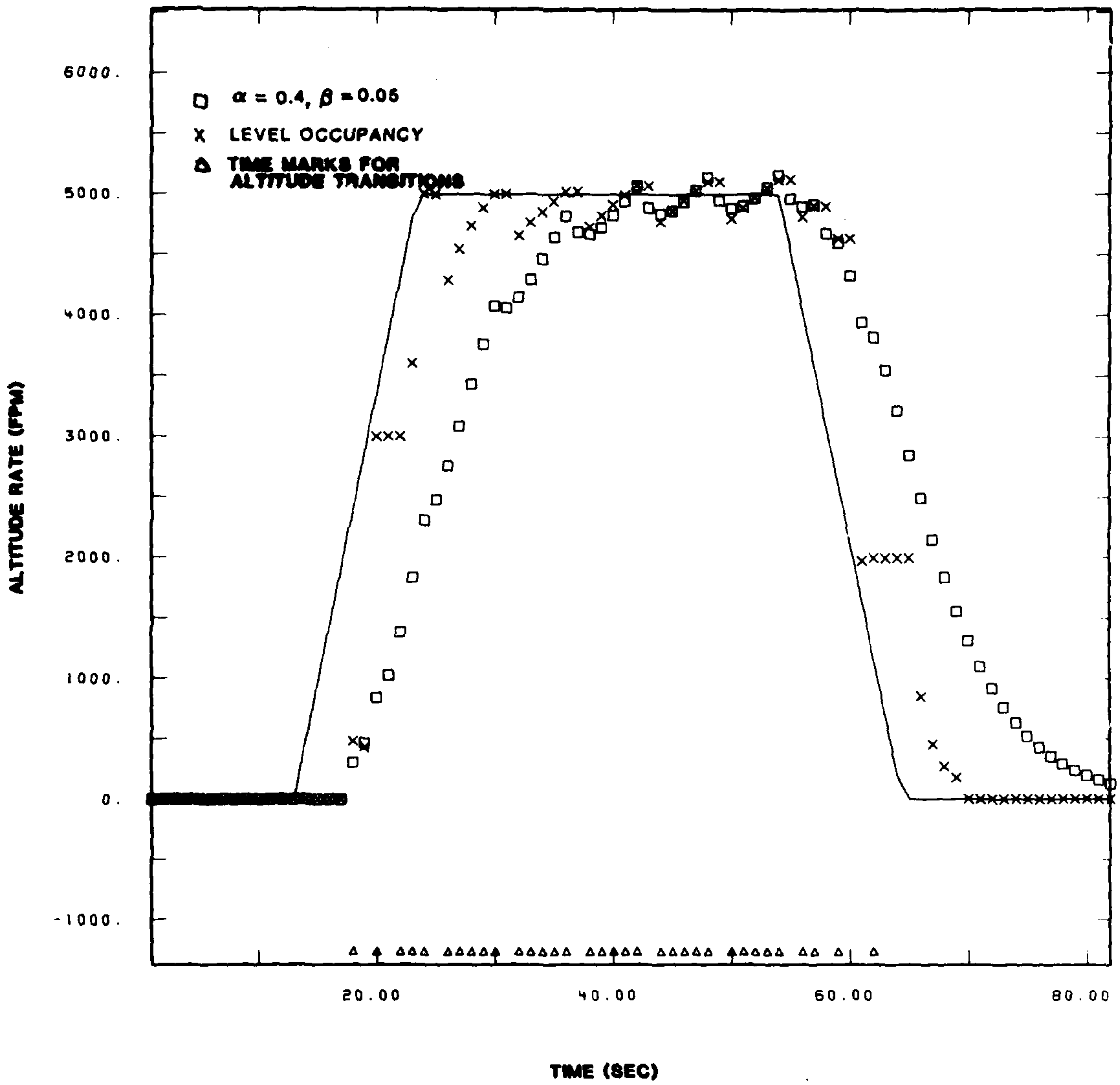

Fig. 7.5. Rate estimation performance for a 5000 FPM ramp profile.

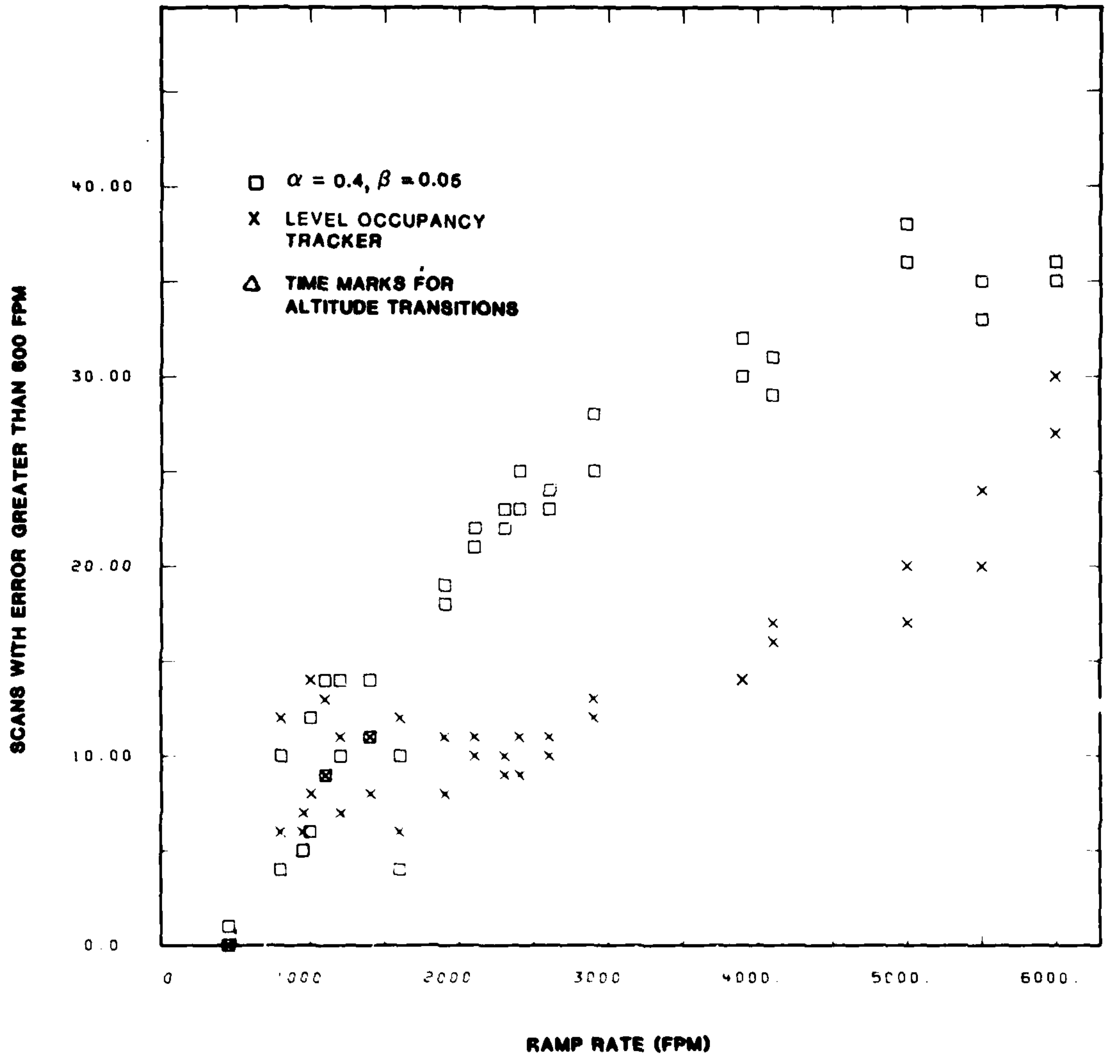

Fig. 7.6. Number of scans with rate error magnitudes greater than 600 FPM for various ramp profiles.

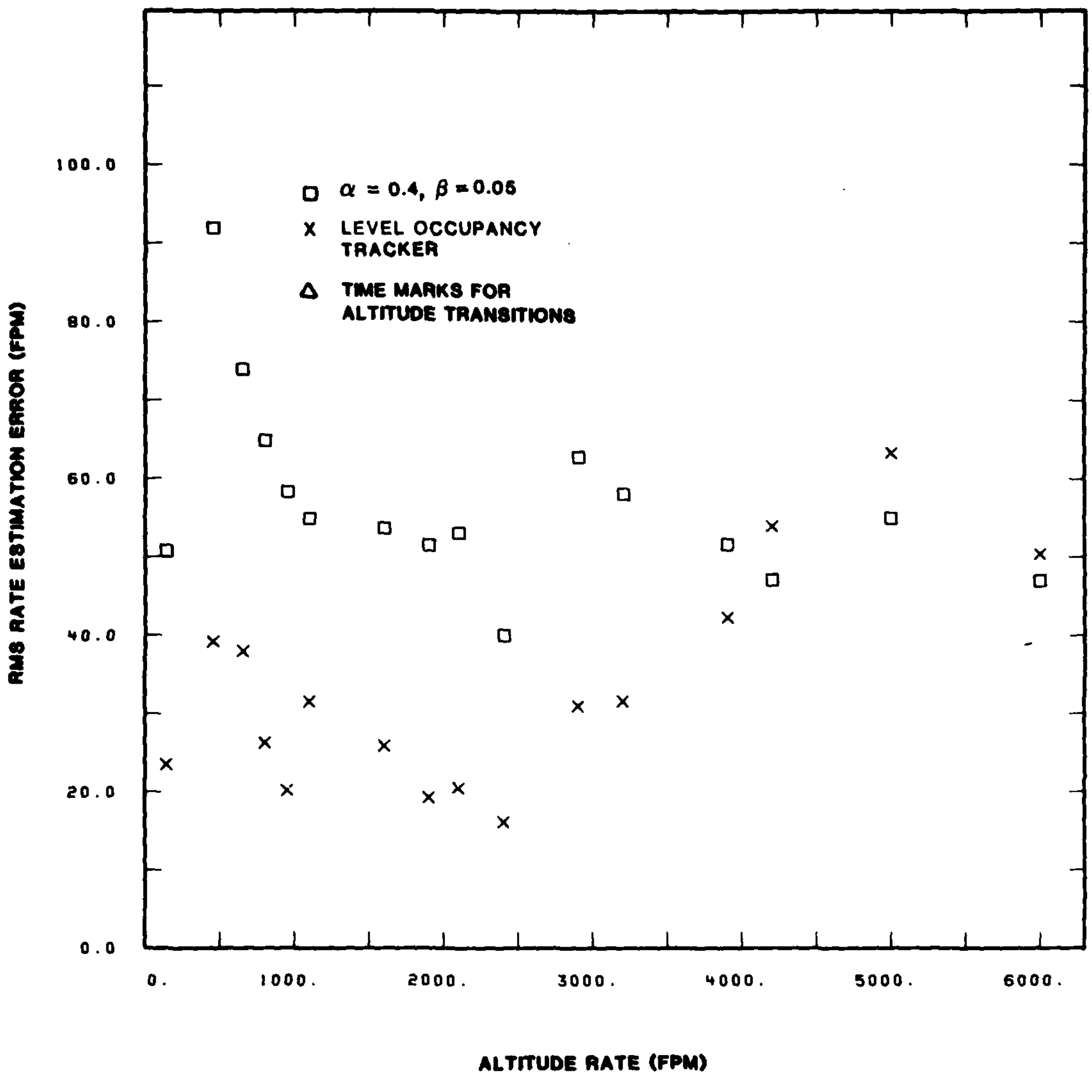

Fig. 7.7. Steady state performance of trackers at 1.0 second update interval.

An algorithm incorporating these changes is provided in Appendix C. The principal logic changes incorporated into this algorithm will be described in the next two subsections. Simulation results for a 4.7 second data rate (typical for terminal air traffic control radars) will then be presented.

8.1 Algorithm Modifications

8.1.1 Generalized Smoothing Equation

The update procedure for estimation of the level occupancy time can be generalized to handle cases in which more than one quantization level is crossed between samples. The basic approach is still to average the level occupancy times. If M transitions have occurred in a single sample interval of duration τ, then the tracker will respond as if M single transitions were reported at intervals of τ/M. Recall that in the averaging process

$$\hat{T}_{n-1} = \frac{\sum_{j=1}^{n-1} T_j^*}{n-1} \qquad (8.1)$$

and for the current update, equation (6.4) is employed with $\beta = 1/n$. Hence for a transition across M levels, the average after update should reflect n-1+M transitions in a time interval which has been incremented by τ over the time of the previous update. Using equation (6.4) this yields

$$\hat{T}_n = \frac{1}{\beta(n-1+M)} [(\hat{T}_{n-1} + \beta(\tau - \hat{T}_{n-1}))], \qquad M>1$$

$$= \frac{1}{1+\beta(M-1)} [(\hat{T}_{n-1} + \beta(\tau - \hat{T}_{n-1}))], \qquad M>1 \qquad (8.2)$$

Note that in the case of M=1, this formula is equivalent to equation 6.4.

8.1.2 Extended Consistency Test

Using the altitude expression provided in equation 3.1, i.e.

$$z(t) = N\,q + \varepsilon_0\,q + \dot{z}\,t,$$

the number of transitions occuring between sample n-1 and sample n can be written

$$M_n = \mathrm{INT}\left[\varepsilon_0 + \frac{n\,\tau}{T}\right] - \mathrm{INT}\left[\varepsilon_0 + \frac{n\,\tau - \tau}{T}\right]$$

Using Theorem 5 of Appendix A, this expression can be written

$$M_n = \begin{cases} \mathrm{INT}\left(\dfrac{\tau}{T}\right) & \text{when } R\left(\varepsilon_0 + \dfrac{n\,\tau}{T}\right) < 1 - R\left(\dfrac{\tau}{T}\right) \\[2ex] \mathrm{INT}\left(\dfrac{\tau}{T}\right) + 1 & \text{when } R\left(\varepsilon_0 + \dfrac{n\,\tau}{T}\right) > 1 - R\left(\dfrac{\tau}{T}\right) \end{cases} \tag{8.3}$$

A consistency test can be based upon M_n in a manner analagous to the consistency test based upon level occupancy times as described in Section 6. Define a residual s_j according to

$$s_j = \frac{\tau}{\hat{T}_{j-1}} - M_j \tag{8.4}$$

This quantity is the expected number of level transitions between samples less the number of transitions which actually occurred. In the absence of acceleration this residual should average to approximately zero and should have maximum magnitude of 1. The residual becomes more positive when there is an acceleration which decreases the rate magnitude and becomes more negative in the presence of acceleration which increases the rate magnitude. A summed value of the residual defined by

$$\bar{s}_j = \gamma\, \bar{s}_{j-1} + s_j$$

can be tested in order to detect accelerations. When $|\bar{s}_j|$ exceeds a selected threshold, the tracker gain is increased to better follow the apparent acceleration.

8.2 Simulation Results at 4.7-Second Update Interval

Appendix C contains a listing for a tracking algorithm modified to function at a 4.7-second update rate according to the principles introduced in Section 8.1. This is a rate that is typical of air traffic control radars. Simulation results for several altitude profiles are provided in Figures 8.1 through 8.4. The parameter values used for the alpha-beta tracker ($\alpha = 0.464$, $\beta = 0.144$) are those employed in the proposed tracker for the Automatic Traffic Advisory and Resolution System (ATARS). It can be seen that although the step function response and tracking cycle behavior are not considerations at the longer update interval, the response to acceleration can still be improved through use of a level occupancy tracker. Figure 8.5 provides a comparison of the error counts of the two trackers for ramp profiles at various rates.

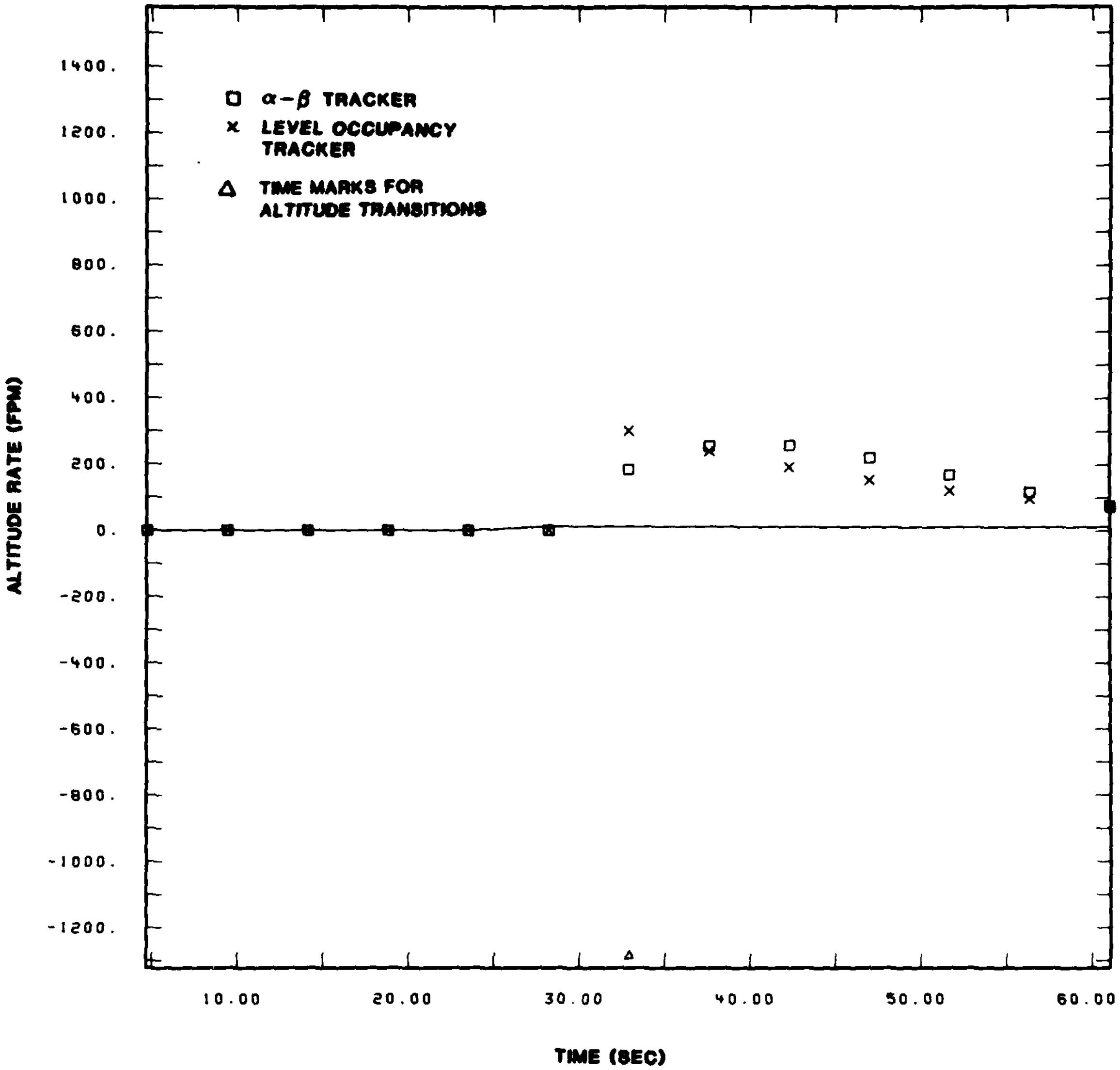

Fig. 8.1. Rate estimation performance for single altitude transition, 4.7 - second update interval.

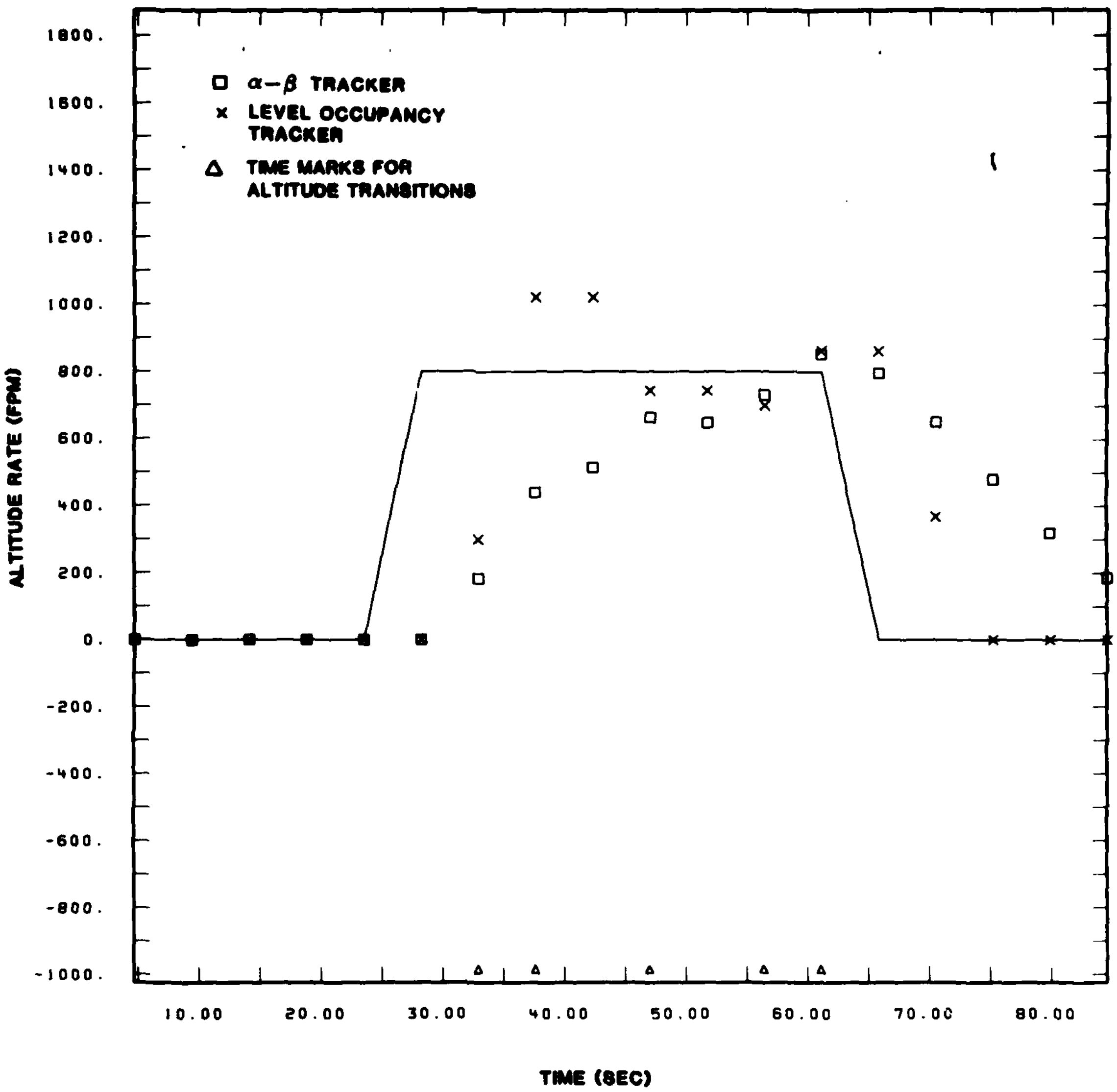

Fig. 8.2. Rate estimation performance for 800 FPM ramp profile, 4.7 - second update interval.

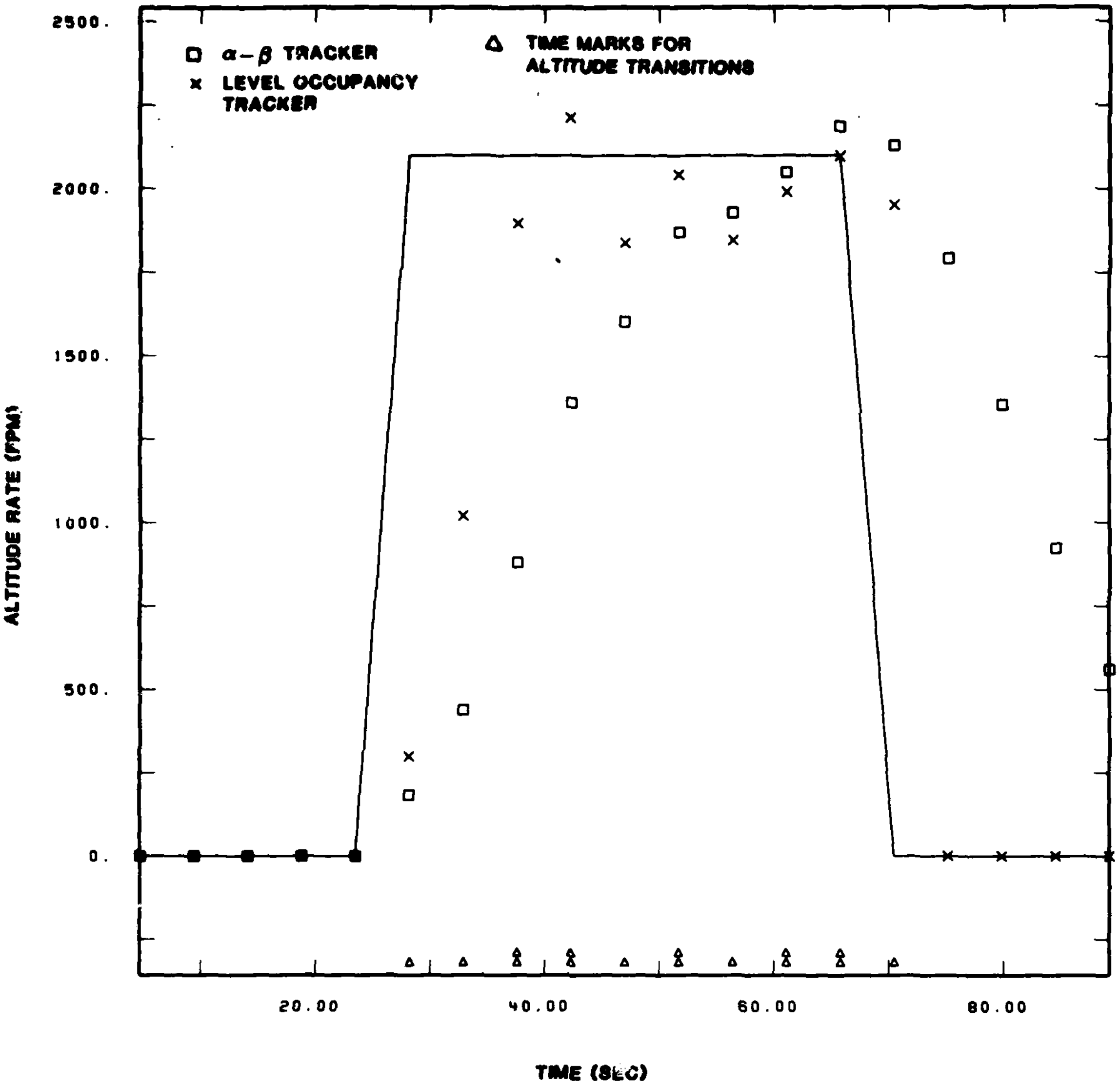

Fig. 8.3. Rate estimation performance for 2100 FPM ramp profile, 4.7 - second update interval.

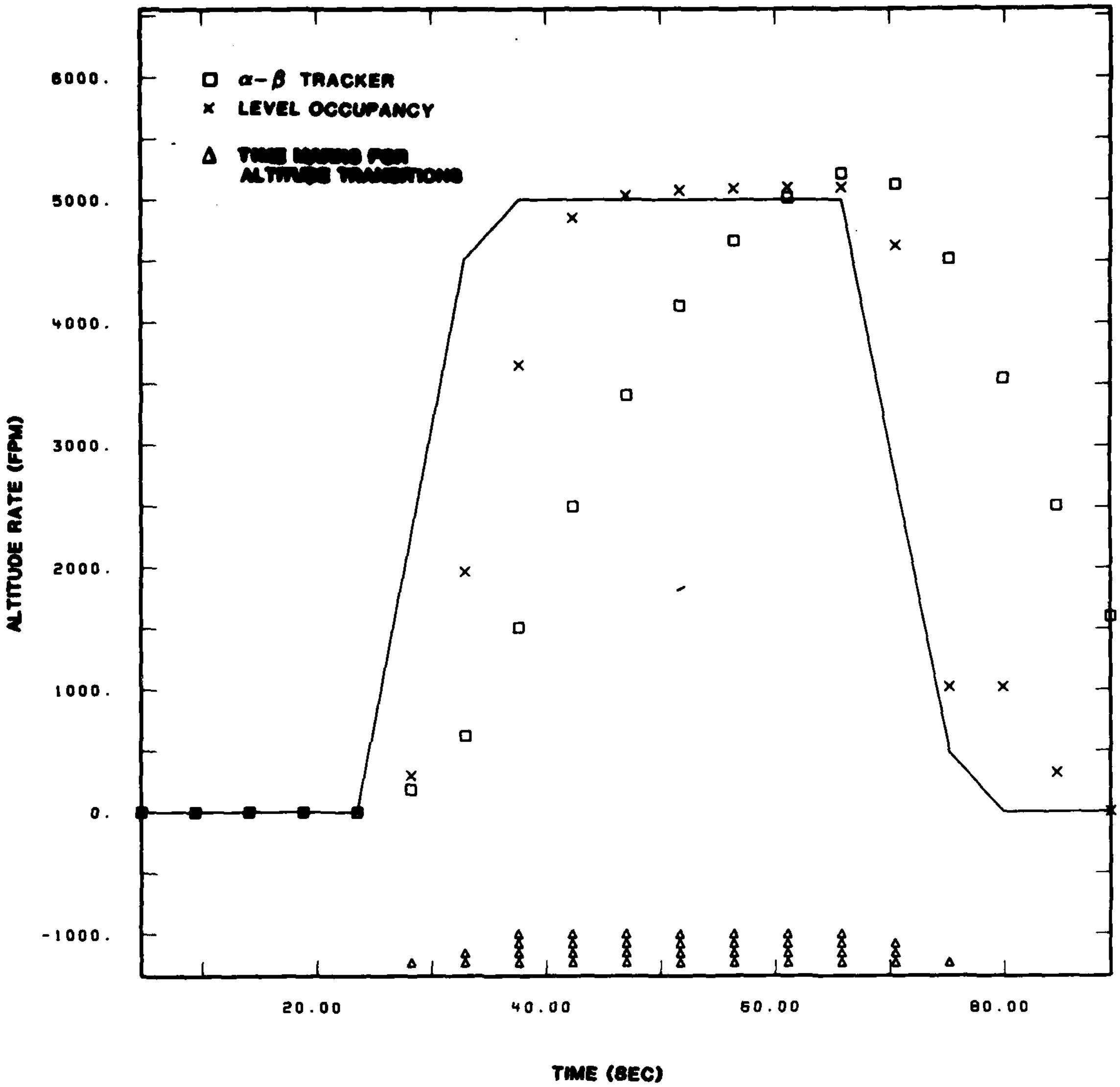

Fig. 8.4. Rate estimation performance for 5000 FPM ramp profile, 4.7 - second update interval.

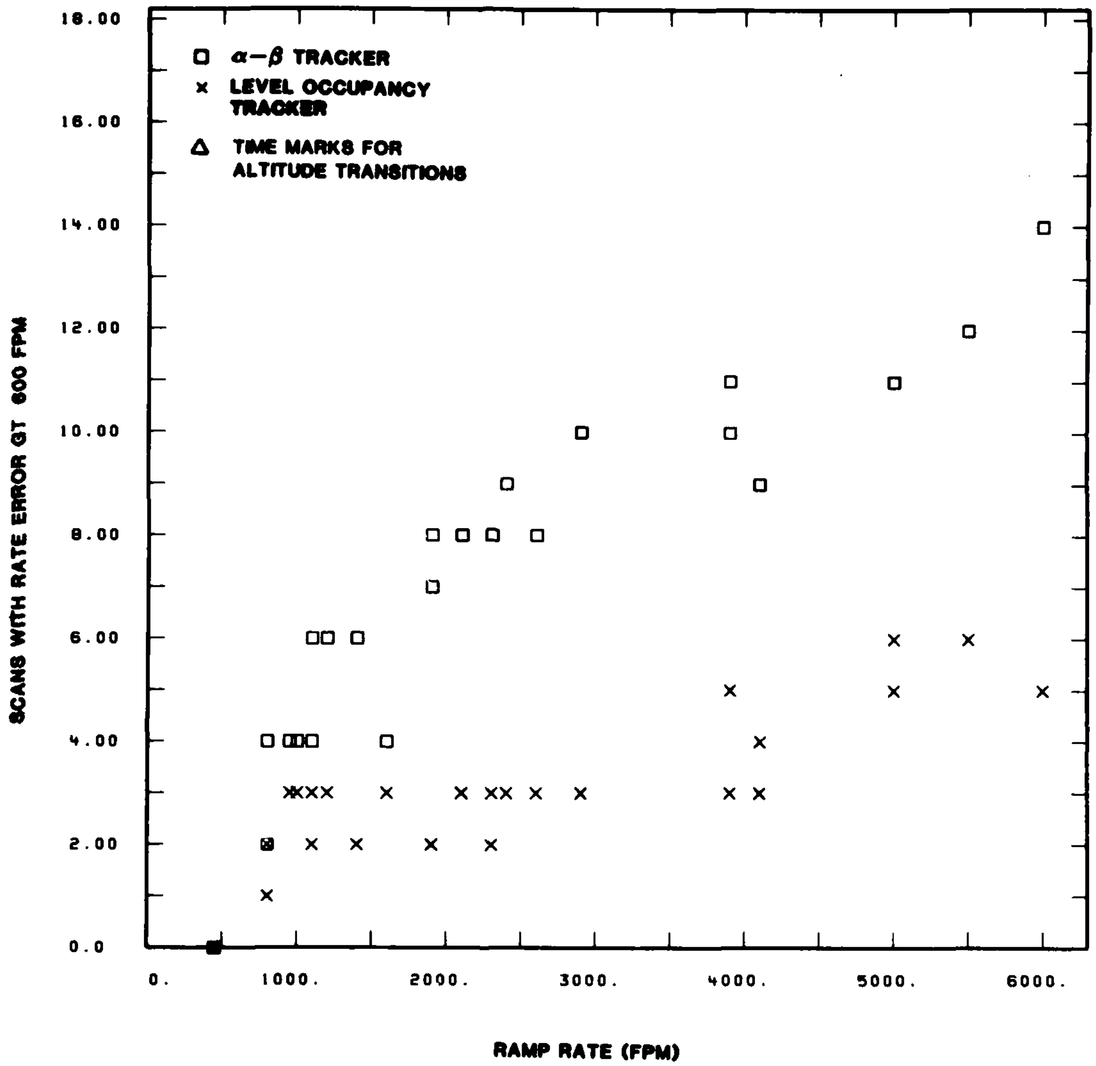

Fig. 8.5. Number of scans on which the rate estimation error exceeded 600 FPM for various ramp profiles for 4.7 - second update interval.

9.0 SUMMARY AND CONCLUSIONS

The simple linear recursive algorithms typically employed in the tracking of aircraft vertical motion exhibit undesirable rate responses following widely spaced transitions between adjacent altitude quantization levels. The selection of tracking gain also involves a difficult trade-off between steady state tracking performance and response to acceleration. An alternative approach to tracking is examined in this document. Instead of smoothing each altitude report in an identical manner, the alternative tracker smooths the time the aircraft spends at each quantization level. This level occupancy tracking results in a rate estimate which is not influenced by redundant samples taken while the aircraft is within the same quantization level. Since two level transitions are necessary to measure level occupancy time, the response to a single isolated transition is arbitrary and is controlled to eliminate problems of over-response in near-level flight. Further improvement in tracking performance results from the use of consistency tests which use properties of quantized data to detect inconsistencies between tracker estimates and observed data. When inconsistencies are discovered, an immediate correction is applied to the estimate and the tracking gain is adjusted accordingly.

Simulation of a particular realization of this type of tracking has been conducted at 1-second and 4.7-second update rates. The simulation indicates that the alternative tracking algorithm can be independently optimized to perform well in cases of isolated transitions, steady state and accelerating trajectories. The most significant performance difference between the conventional alpha-beta tracking algorithms and the alternative algorithm is in the response to the initiation or termination of altitude rates above 1500 FPM magnitude.

APPENDIX A

MATHEMATICAL RELATIONSHIPS

This appendix presents certain mathematical definitions and relationships which are useful in the analysis of quantization effects. In the following equations the quantities m and n are integers.

__Definition:__ $INT(x)$ = the largest integer less than or equal to x.

__Examples:__ $INT(3.6) = 3.0$, $INT(3.0) = 3.0$, $INT(-.6) = -1.0$.

__Definition:__ $R(x) = x - INT(x)$

__Examples:__ $R(3.6) = 0.6$, $R(3.0) = 0.$, $R(-.6) = 0.4$
Note that for $x \geq 0$, $R(x)$ is the fractional part of x.

__Theorem 1:__ $0 \leq R(x) < 1$

__Theorem 2:__ $INT(m + x) = m + INT(x)$

__Theorem 3:__ $R(n + x) = R(x)$

__Theorem 4:__ $R(n - x) = R[1-R(x)]$

__Theorem 5:__

$$INT(x + y) - INT(y) = \begin{cases} INT(x) & \text{when } 1 - R(x) > R(y) \\ INT(x) + 1 & \text{when } 1 - R(x) \leq R(y) \end{cases}$$

__Theorem 6:__

$$R(x + y) = R(x) + R(y) - INT\,[R(x) + R(y)]$$

or equivalently

$$R(x + y) = \begin{cases} R(x) + R(y) & \text{when } R(x) + R(y) < 1 \\ R(x) + R(y) - 1 & \text{when } R(x) + R(y) \geq 1 \end{cases}$$

APPENDIX B

ALGORITHM FOR 1-SECOND UPDATE INTERVAL

This appendix provides details on the level occupancy tracking algorithm used to generate the simulation results for a 1.0 second update rate (see Section 7.0). An overview of the principal sections of the logic is provided in Figure B.1. Tables B.1 and B.2 provide definitions of variables in the track file and parameters used in the logic. The logic was divided into two FORTRAN subroutines. The first subroutine (see Figure B.2) initializes new tracks. In BCAS a track initiated by the BCAS surveillance function exists prior to activation of the collision avoidance logic and hence this track is used for initialization. But the simple surveillance tracker does not store all the quantities needed to fully initialize the level occupancy tracker. In particular, it does not store the time of the preceding altitude transition. Hence those portions of the logic which test level occupancy must be by-passed until either the second altitude transition is observed or enough time (about 18 seconds) has passed to guarantee that the aircraft is in near level flight. During the interim period, the "start-up" logic employs simple alpha-beta smoothing equations.

A FORTRAN listing for the update logic is provided in Figure B.3. The following notes will be helpful in implementation of this algorithm:

1) The quantities Q and DT are treated as parameters in the software, but they can be replaced by their fixed values (Q = 100 feet and DT = 1.0 second).

2) As currently written, the state vector elements ZMOD(8) and ZMOD(9) are not used simultaneously. Hence it would be possible to revise the code to treat them as the same quantity, thus reducing the size of the aircraft state vector by one element.

3) The quantity ZMOD(6) is the time of last track update. If the track is always updated at a specified rate, this quantity need not be stored. This would reduce the size of the aircraft state vector by one element.

4) The following quantities are internal to the subroutines and are defined for computational convenience.

BETA1	ISGN
BLIM	QSIGN
DBINS	TCUR
DELT	TEST
DZM	TNDEX

5) In the code presented here an altitude report of zero is used to indicate missing data. Actually, zero is a valid Mode-C report value. In implementing this code, another default value should be chosen.

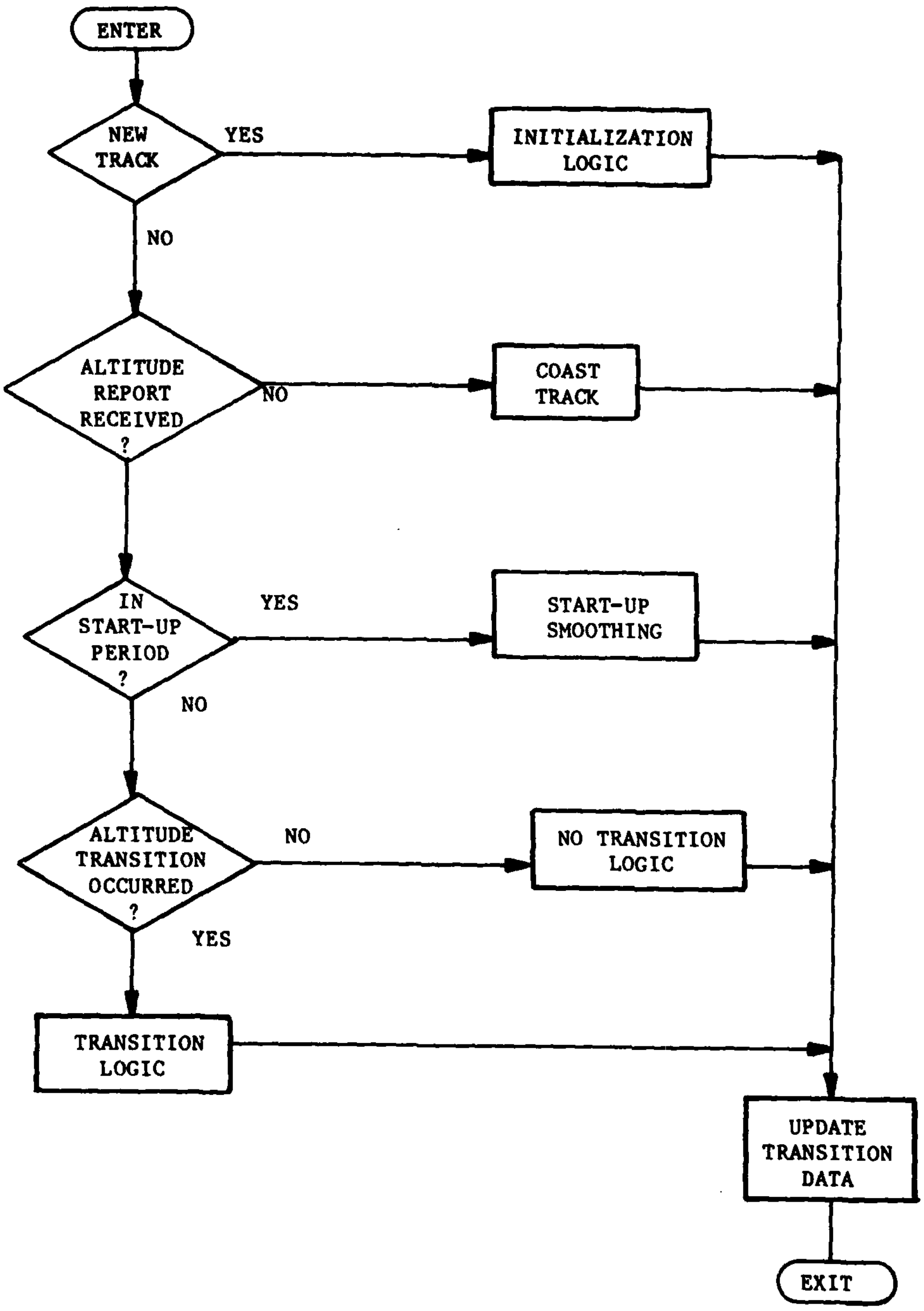

Fig. B.1. Overview of tracking algorithm showing principal modules.

TABLE B.1

CONTENTS OF THE TRACK FILE

ZMOD(1)	Estimated Altitude (ft)
ZMOD(2)	Estimated altitude rate (fps)
ZMOD(3)	Time last Mode-C report was received (sec)
ZMOD(4)	Previously reported altitude (ft)
ZMOD(5)	Time of transition to previously reported altitude (sec)
ZMOD(6)	Time of last track update (sec)
ZMOD(7)	Estimated level occupancy time (sec)
ZMOD(8)	Firmness of rate. If equal to zero, indicates rate is based upon assumption of level flight or observation of a single altitude transition. If equal to 1 or more, it equals the number of observed occupancy times for the current rate (but it may be reset by consistency tests).
ZMOD(9)	Start-up Counter. Used in establishing track.
ZMOD(10)	Summed residual. Used to detect a trend in the tracker residuals (which indicates vertical acceleration).
ZM	Mode-C altitude report. Set to 0 when no report has been received.

TABLE B.2

PARAMETERS USED IN ALTITUDE TRACKING

VARIABLE NAME	DEFINITION	NOMINAL VALUE
DT	Nominal Time Between Updates	1.0 sec. (BCAS)
Q	Quantization Bin Width	100 ft.
P1	Magnitude of Rate Allowed Following Isolated Altitude Transition	8 FPS
P3	Decay Factor When No Reinforcing Transition has Occurred	0.90
P4	Stiff Rate Smoothing Parameter	0.04
P5	Excess Bin Occupancy Time Which Results in Transition to Level Flight	5.0 sec.
P6	Excess Bin Occupancy Time Which Results in Correction to Altitude Rate (units of DT)	1.5
P7	Amount of Discrepancy in Bin Occupancy Times Which Triggers Reinitialization of Tracker Vertical Rate (units of DT)	1.5
P8	Parameter Used to Position an Estimated Bin Transition Time Within an Interval of Missing Data	0.6
P9	Position Smoothing Parameter	0.3
P10	Smoothing Gain Used To Compute Summed Residual, ZMOD(10)	0.80
P11	Value of Bin Occupancy Smoothing Parameter Used When Excess Residuals are Detected	0.70
P12	Reset Magnitude for the Summed Residual, ZMOD(10)	0.30
P13	Value of ZMOD(9) at Which Transition From Start-up Smoothing to Normal Smoothing Occurs	18
P14	Threshold Magnitude for ZMOD(10). Used to Detect Excess Summed Residual	1.35

ENTER

```
ZMOD (1)  = ZAB(1)
ZMOD (2)  = ZAB(2)
ZMOD (3)  = T
ZMOD (4)  = ZM
ZMOD (6)  = ZMOD(3)
ZMOD (7)  = Q/|ZMOD(2)|
ZMOD (8)  = 5.
ZMOD (9)  = 0.
ZMOD (10) = 0.
ZMOD (5)  = -ZMOD(7) + 7*DT
```

EXIT

Fig. B.2. Logic used to initialize a track based upon a previous existing track with altitude ZAB(1) and altitude rate ZAB(2). T is current time and ZM is the currently reported altitude.

```
      SUBROUTINE VTRMOD
C - - - - MODIFIED VERTICAL TRACKING           11-19-80  JWA
      COMMON/COMA/ALFA,BETA,T,DT,ZM,ZAC(6),ZAB(6),
     K RMS(4),IOUT,KDEV,IDTRAJ,KTITLE(10),DIAG(20)
      COMMON/COMMOD/PARM,DTM,Q,ZDINIT,ZMOD(12),ALFA1,BETA1,
     K IPATH(20),KPATH(20),P1,P2,P3,P4,P5,P6,P7,P8,
     K P9,P10,P11,P12,P13,P14
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      IF (T.GT.1.) GO TO 202
C - - - - - - - - - - - - - - - - - - - - - PARAMETER SETTING - - - - - - -
           P1=8.
           P3=0.90
           P4=.04
           P5=5.
           P6=1.5
           P7=1.5
           P8=.6
           P9=.3
           P10=.80
           P11=.7
           P12=.3
           P13=18.
           P14=1.35
202   CONTINUE
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - UPDATE - - - - - - - -
      ZP=ZMOD(1)+(T-ZMOD(6))*ZMOD(2)
      IF (ZM.GT.0.) GO TO 490
C - - - - - - - - - - - - - - - ALTITUDE REPORT MISSING: COAST TRACK - - - -
           ZMOD(1)=ZP
      GO TO 802
490   CONTINUE
      DZM=ZM-ZMOD(4)
      DBINS=ABS(DZM)/Q
      ZMOD(9)=ZMOD(9)+1.+10*DBINS
      IF (ZMOD(9).GT.P13) GO TO 495
C - - - - - - - - - - - - - - - - - - - - - - START-UP SMOOTHING - - - - - - - - -
           ZMOD(1)=ZP+P9*(ZM-ZP)
           ZMOD(2)=ZMOD(2)+P4*(ZM-ZP)/(T-ZMOD(6))
           IF (DZM.NE.0.) GO TO 590
           GO TO 802
495   CONTINUE
      IF (DZM.EQ.0.) GO TO 701
C - - - - - - - - - - - - - - - - - - - - - - - - - - -TRANSITION LOGIC - - - - - - -
      ISGN=INT(SIGN(1.,DZM))
      QSIGN=Q*ISGN
      TEST=ZMOD(2)*DZM
      IF (TEST.LE.100.) GO TO 580
C - - - - - - - - - - - - - - - - - - - - - - - TREND EXISTS - - - - - - - -
      TPREV=(T-ZMOD(5))/DBINS
      DELT=TPREV-ZMOD(7)
```

Fig. B.3. FORTRAN subroutine used to update a track at 1 second intervals. This routine is called once each scan following the scan of initialization. As a programing convenience, the setting of parameters has been incorporated into this sub routine.

```
      IF (ZMOD(8).LE.0.) GO TO 540
      IF (ABS(DELT).GT.P7*DT) GO TO 540
      ZMOD(10)=P10*ZMOD(10)+DELT
      DIAG(1)=ZMOD(10)
      IF (ABS(ZMOD(10)).LE.P14) GO TO 530
C - - - - - - - - - - - - - - - - - - - - - - EXCESS RESIDUAL DETECTED- - -
           BETA1=P11
           ZMOD(8)=3.
           ZMOD(10)=SIGN(P12,ZMOD(10))
           GO TO 533
530   CONTINUE
C - - - - - - - - - - - - - - - - - - CURRENT RATE WITHIN LIMITS : SMOOTH - -
      Z7=ZMOD(7)
      BLIM=(Z7-1.)**2/(Z7**2+64.)
      BETA1=AMAX1(1./(ZMOD(8)+1.),BLIM,.08)
      ZMOD(8)=ZMOD(8)+1.
533   CONTINUE
      ZMOD(7)=ZMOD(7)+BETA1*(TPREV-ZMOD(7))
      ZMOD(2)=QSIGN/ZMOD(7)
      ZMOD(1)=ZP+P9*(ZM-ZP)
      GO TO 590
540   CONTINUE
C - - - - - - - - - - - - - - - - - - - - - - REINITIALIZE RATE - - - - - -
      ZMOD(7)=AMAX1(TPREV,1.4)
      ZMOD(2)=QSIGN/ZMOD(7)
      ZMOD(10)=0.
      ZMOD(8)=1.
      ZMOD(1)=ZM-QSIGN/2+ZMOD(2)*DT/2
      GO TO 590
580   CONTINUE
C - - - - - - - - - - - - SET RATE TO VALUE FOR SINGLE TRANSITION -
      ZMOD(2)=P1*ISGN
      ZMOD(1)=ZM-QSIGN/2+ZMOD(2)*DT/2
      ZMOD(7)=QSIGN/ZMOD(2)
      ZMOD(8)=0.
      ZMOD(10)=0.
590   CONTINUE
C - - - - - - - - - - - - - - - UPDATE LEVEL TRANSITION DATA - - - - - -
      ZMOD(4)=ZM
      ZMOD(5)=T
      IF (ZMOD(3).GE.ZMOD(6)) GO TO 802
C - - - CORRECT TRANSITION TO FALL WITHIN PERIOD OF MISSING DATA - - -
      ZMOD(5)=T+P8*(ZMOD(3)-T+DT)
      GO TO 802
C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C - - - - - - - - - - - - - - - - - - - NO TRANSITION LOGIC - - - - - - -
701   CONTINUE
      ZMOD(1)=ZP+P9*(ZM-ZP)
      TCUR=T-ZMOD(5)+DT
      TNDEX=(TCUR-ZMOD(7))/DT
```

Fig. B.3. Continued.

```
      IF (TNDEX.GT.P5) GO TO 630
      IF (TNDEX.GE.P6) GO TO 610
C - - - - - - - - - - - - NORMAL UPDATE - NO TRANSITION EXPECTED - - - - -
C - - - - - - - - - - - - - - - - - - RATE DECAYS ACCORDING TO P3 - - - - -
      IF (ZMOD(8).GE.1.) GO TO 802
      ZMOD(2)=ZMOD(2)*P3
      ZMOD(7)=Q/(ABS(ZMOD(2))+.1)
      GO TO 802
610   CONTINUE
C - - - - BIN OCCUPANCY LONGER THAN EXPECTED - ALTER EXTERNAL RATE - - -
      T7=ZMOD(7)+(.3*ZMOD(7)+.5*DT)*(TNDEX-.3)**2
      ZMOD(2)=SIGN(Q,ZMOD(2))/T7
      ZMOD(8)=AMAX1(2.,ZMOD(8)-1.)
      GO TO 802
630   CONTINUE
C - - - - - - - - - - - - - - - - - - - - TRANSITION TO LEVEL FLIGHT - - -
      ZMOD(1)=ZM
      ZMOD(2)=0.
      ZMOD(7)=99.
      ZMOD(8)=0.
      ZMOD(10)=0.
802   CONTINUE
      ZMOD(8)=AMIN1(ZMOD(8),10.)
      IF (ZM.GT.0.) ZMOD(3)=T
      ZMOD(6)=T
      RETURN
      END
```

Fig. B.3. Continued.

APPENDIX C

ALGORITHM FOR 4.7-SECOND UPDATE INTERVAL

A FORTRAN subroutine used to implement a level occupancy tracker for a 4.7 second update interval is given in Figure C.1. Variable definitions and further explanation of the algorithmic structure can be found in Appendix B and Section 8.0 of this document.

```
      SUBROUTINE VTRMOD
C - - - - MODIFIED VERTICAL TRACKING FOR 4.7 SECONDS UPDATE RATE
      COMMON/COMA/ALFA,BETA,T,DT,ZM,ZAC(6),ZAB(6),
     K RMS(4),IOUT,KDEV,IDTRAJ,KTITLE(10),DIAG(20)
      COMMON/COMMOD/PARM,DTM,Q,ZDINIT,ZMOD(12),ALFA1,BETA1,
     K IPATH(20),KPATH(20),P1,P2,P3,P4,P5,P6,P7,P8,
     K P9,P10,P11,P12,P13,P14
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      IF (ZMOD(9).GE.1.) GO TO 202
C - - - - - - - - - - - - - - - - - - - - - - PARAMETER SETTING - - - - - - -
           P1=5.
           P2=0.6
           P3=0.80
           P4=.10
           P5=2.5
           P6=0.9
           P7=1.5
           P8=.6
           P9=.3
           P10=.50
           P11=.7
           P12=0.2
           P13=22.
           P14=1.3
202   CONTINUE
      ZP=ZMOD(1)+(T-ZMOD(6))*ZMOD(2)
      IF (ZM.GT.0.) GO TO 490
C - - - - - - - - - - - - - - - - ALTITUDE REPORT MISSING: COAST TRACK - - - -
           ZMOD(1)=ZP
C - - - - - - TO BE ADDED: IF  T-ZMOD(3) GT THRESHOLD, DROP TRACK - - - -
      GO TO 802
490   CONTINUE
      DZM=ZM-ZMOD(4)
      DBINS=ABS(DZM)/Q
      ZMOD(9)=ZMOD(9)+4.
      IF (ZMOD(9).GT.P13) GO TO 495
C - - - - - - - - - - - - - - - - - - - - -START-UP SMOOTHING - - - - - - - -
           ZMOD(1)=ZP+P9*(ZM-ZP)
           ZMOD(2)=ZMOD(2)+P4*(ZM-ZP)/(T-ZMOD(6))
           IF (DZM.NE.0.) GO TO 590
           GO TO 802
495   CONTINUE
      IF (DZM.EQ.0.) GO TO 701
C - - - - - - - - - - - - - - - - - - - - - - - - - -ALT REPORT CHANGED - -
      ISGN=INT(SIGN(1.,DZM))
      QSIGN=Q*ISGN
      TEST=ZMOD(2)*DZM
      IF (TEST.LE.100.AND.DBINS.EQ.1.) GO TO 580
C - - - - - - - - - - - - - - - - - - - TREND EXISTS - - - - - - -
      TPREV=(T-ZMOD(5))/DBINS
```

Fig. C.1. FORTRAN subroutine used to update a track at 4.7 second intervals. This routine is called once each scan following the scan of initialization. As a programing convenience, the setting of parameters has been incorporated into this subroutine.

```
C - - - - - - - - - - - - - - - - - NO TRANSITION LOGIC - - - - - - - - - - -
701   CONTINUE
      ZMOD(1)=ZP+P9*(ZM-ZP)
      TCUR=T-ZMOD(5)+DT
      TNDEX=(TCUR-ZMOD(7))/DT
      IF (ZMOD(7).LT.0.8*DT) TNDEX=TCUR/ZMOD(7)
      IF (TNDEX.GT.P5) GO TO 630
      IF (TNDEX.GE.P6) GO TO 610
C - - - - - - - - - - - - - NORMAL UPDATE - NO TRANSITION EXPECTED - - - - -
C - - - - - - - - - - - - - -RATE DECAYS ACCORDING TO P3 - - - - -
      IF (ZMOD(8).GE.1.) GO TO 802
      ZMOD(2)=ZMOD(2)*P3
      ZMOD(7)=Q/(ABS(ZMOD(2))+.1)
      GO TO 802
610   CONTINUE
C - - - - BIN OCCUPANCY LONGER THAN EXPECTED - ALTER EXTERNAL RATE - - -
      ZMOD(2)=SIGN(Q,ZMOD(2))/(ZMOD(7)+(.4*ZMOD(7)+1.0*DT)*
     K (TNDEX-0.4)**2)
      ZMOD(8)=AMAX1(2.,ZMOD(8)-1.)
      GO TO 802
630   CONTINUE
C - - - - - - - - - - - - - - - - - - TRANSITION TO LEVEL FLIGHT - - -
      ZMOD(1)=ZM
      ZMOD(2)=0.
      ZMOD(7)=99.
      ZMOD(8)=0.
      ZMOD(10)=0.
802   CONTINUE
      IF (ZM.GT.0.) ZMOD(3)=T
      ZMOD(6)=T
      RETURN
      END
```

Fig. C.1. Continued.

```
      DELT=TPREV-ZMOD(7)
      DZ10=DELT/DT
      IF (ZMOD(7).LT.DT) DZ10=1.*(DT/ZMOD(7)-DBINS)
      IF (ZMOD(8).LE.0.) GO TO 540
      IF (ABS(DZ10).GT.P7) GO TO 540
      ZMOD(10)=P10*ZMOD(10)+DZ10
      IF (ABS(ZMOD(10)).LE.P14) GO TO 530
C - - - - - - - - - - - - - - - - EXCESS RESIDUAL DETECTED- - - - - - - -
            BETA1=P11
            ZMOD(8)=2.
            ZMOD(10)=SIGN(P12,ZMOD(10))
            GO TO 533
530   CONTINUE
C - - - - - - - CURRENT RATE WITHIN LIMITS : SMOOTH - - - - - - - -
      Z7=ZMOD(7)
      BLIM=(Z7-1.)**2/(Z7**2+64.)
      BETA1=AMAX1(1./(ZMOD(8)+.6),BLIM,.10)
      ZMOD(8)=AMIN1(ZMOD(8)+1.,10.)
533   CONTINUE
      ZMOD(7)=(ZMOD(7)+BETA1*(DBINS*TPREV-ZMOD(7)))/(1.+BETA1*(DBINS-1.)
     K  )
      ZMOD(2)=QSIGN/ZMOD(7)
      ZMOD(1)=ZP+P9*(ZM-ZP)
      GO TO 590
540   CONTINUE
C - - - - - - - - - - - - - - - - - - REINITIALIZE RATE - - - - - --
      Z7PREV=ZMOD(7)
      ZMOD(7)=1.2*TPREV+.05*DT
      ZMOD(2)=QSIGN/ZMOD(7)
      ZMOD(10)=0.
      ZMOD(8)=1.
      ZMOD(1)=ZM-QSIGN/2+ZMOD(2)*DT/2
      GO TO 590
580   CONTINUE
C - - - - - - - - - - - - SET TO A PRIORI VALUE FOR SINGLE TRANSITION -
      ZMOD(2)=P1*ISGN
      ZMOD(1)=ZM-QSIGN/2+ZMOD(2)*DT/2
      ZMOD(7)=QSIGN/ZMOD(2)
      ZMOD(8)=0.
      ZMOD(10)=0.
590   CONTINUE
C - - - - - - - - - - - - - - UPDATE LEVEL TRANSITION DATA - - - - - - -
      ZMOD(4)=ZM
      ZMOD(5)=T
      ZMOD(9)=ZMOD(9)+10.
      IF (ZMOD(3).GE.ZMOD(6)) GO TO 802
C - - - - CORRECT TRANSITION TO FALL WITHIN PERIOD OF MISSING DATA - - -
      ZMOD(5)=T+P8*(ZMOD(3)-T+DT)
      GO TO 802
C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Fig. C.1. Continued.